



Marco Aurélio Marques Correia

Licenciatura em Ciências da Engenharia Mecânica

Optimização Robusta de Estruturas

Dissertação para obtenção do Grau de Mestre em
Engenharia Mecânica

Orientador: Doutor João Mário Burgete Botelho Cardoso,
Prof. Auxiliar, FCT-UNL

Júri:

Presidente: Prof. Doutor Pedro Samuel Gonçalves Coelho

Arguente: Prof.^a Doutora Marta Isabel Pimenta Verdete da Silva Carvalho

Vogais: Prof. Doutor João Mário Burgete Botelho Cardoso



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Optimização Robusta de Estruturas

Nota: Esta dissertação não foi escrita ao abrigo do novo acordo ortográfico de 1990.

Copyright @ Marco Aurélio Marques Correia, FCT /UNL, UNL

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimentos

Em primeiro lugar, agradeço ao Professor João Cardoso, por me dar o apoio e orientação necessários à elaboração e conclusão desta dissertação. Pela partilha de conhecimentos, a disponibilidade sempre manifestada e todo o empenho que imprimiu neste projecto. Agradeço também pelo tempo dispendido, pela confiança depositada e pela simpatia e calma com que me guiou nesta importante etapa da minha formação.

A todos os meus colegas de faculdade com quem convivi, trabalhei e aprendi e que, ao longo destes anos, me apoiaram a alcançar objectivos. Um Obrigado a todos eles e em particular ao Júlio Balagões.

Desejo agradecer a todos os meus amigos que me acompanharam ao longo dos vários anos de formação, sempre presentes nos bons e maus momentos e que sempre me deram palavras de incentivo, em particular ao Mário Marinheiro e João Cruz.

À Joana, o meu profundo agradecimento pela paciência, compreensão, apoio e ajuda prestados durante todo o desenvolvimento deste trabalho.

Por fim, mas não menos importante, agradeço à minha família, em particular aos meus pais, pelo apoio incansável e incondicional. Sem a sua ajuda não seria possível e, por isso, dedico-lhes este trabalho.

À minha irmã, sobrinho e afilhado e aos meus avós, um grande obrigado por tornarem mais tranquila a concretização deste projecto, com o seu apoio e carinho.

Resumo

A otimização com constrangimentos de fiabilidade (RBDO) procura obter sistemas estruturais seguros e otimizados, considerando a variabilidade dos parâmetros relacionados com propriedades do material, cargas aplicadas e geometria. A otimização robusta (RDO) tenta encontrar esses sistemas que, para além disso, apresentam menor variabilidade da resposta estrutural.

O objectivo desta dissertação é estudar problemas de RBDO e de RDO, aplicando novas formulações para estes últimos. Também se pretende criar uma plataforma básica para ser usada e desenvolvida por estudantes de engenharia estrutural e por investigadores, a fim de promover a aplicação destas técnicas de optimização. Esta plataforma é constituída por um conjunto de programas, escritos em MATLAB, capazes de resolver estes dois tipos de problemas. É composta por, um programa de análise estrutural capaz de calcular sensibilidades utilizando o método contínuo, um programa de análise de fiabilidade que utiliza o método FORM e um algoritmo de optimização (SQP) que está implementado na *optimtool* do MATLAB.

A eficiência do sistema proposto é demonstrada através da resolução de problemas RBDO e RDO; para isso são apresentados quatro exemplos onde são utilizadas todas as funcionalidades dos programas desenvolvidos.

Palavras Chave: Análise de fiabilidade estrutural, Optimização com constrangimentos de fiabilidade, Optimização robusta.

Abstract

Reliability-based design optimization (RBDO) searches for safe and optimized structural systems regarding variability of parameters related to material properties, loading or geometry. Robust design optimization (RDO) tries to find those systems that also have minimal variability of structural response.

The objective of this dissertation is to study different RDO formulation and also to provide a basic platform to be used and further developed by structural engineering students and researchers, in order to promote the application of these optimization techniques. This thesis presents a set of MATLAB programs designed to solve these two types of problems. This set consists of, a structural analysis program capable of computing sensitivity using the continuum method, a reliability analysis program that uses the first order reliability method (FORM) and an optimization algorithm, (sequential quadratic programming - SQP), implemented in the MATLAB *optimtool* Toolbox.

The efficiency of the proposed system is demonstrated by the solving of RBDO and RDO problems; for this purpose four examples are presented where all the features of the developed programs are used.

Keywords: Structural reliability analysis, Reliability-based design optimization, Robust design optimization.

Índice Geral

Agradecimentos	III
Resumo	V
Abstract	VII
Siglas	XV
Capítulo 1 - Introdução.....	1
1.1 Motivação e Enquadramento.....	1
1.2 Objectivos	4
1.3 Estrutura da Dissertação.....	5
Capítulo 2 - Fiabilidade estrutural	7
2.1 Introdução	7
2.1.1 Incerteza	7
2.1.2 Variáveis aleatórias e distribuição probabilísticas	8
2.1.3 Estados limite	9
2.2 Análise de Fiabilidade de estruturas	11
2.3 Métodos de Integração e Simulação	16
2.3.1 Algoritmo FORM.....	17
Capítulo 3 - Análise de Sensibilidades.....	25
3.1 Introdução	25
3.2 Análise de Sensibilidades – método contínuo	26
3.3 Método das diferenças finitas	33
3.4 Exemplo	34
Capítulo 4 - Optimização Robusta.....	37
4.1 Optimização Estrutural.....	37
4.2 Optimização com constrangimentos de fiabilidade - RBDO	38
4.3 Optimização Robusta - RDO	40
Capítulo 5 - Programa PROAES	45
5.1 Introdução	45
5.1.1 Constituição dos programas	46

5.2 Funcionamento do programa.....	47
5.2.1 Análise de elementos finitos e/ou sensibilidades	47
5.2.2 Análise de fiabilidade de estruturas pelo método FORM	49
5.2.3 Optimização RBDO e RDO	50
Capítulo 6 - Exemplos	53
6.1 Introdução	53
6.2 Exemplo 1 – Pórtico	54
6.3 Exemplo 2 – Pórtico de 2 andares	57
6.4 Exemplo 3 – Treliça de 31 barras	65
6.4.1 RBDO	67
6.4.2 RDO	70
6.5 Exemplo 4 – Plataforma Offshore	76
6.5.1 RBDO	82
6.5.2 RDO	83
6.6 Conclusões	85
Capítulo 7 - Conclusões.....	87
Bibliografia.....	89
Anexos.....	95

Índice de Figuras

Figura 2.1 - Fontes de incerteza [9].....	8
Figura 2.2 - Representação genérica da função $g(x)$ e da fronteira entre as regiões de segurança e falha para um problema com 2 variáveis aleatórias [15].....	10
Figura 2.3 - Representação das variáveis aleatórias, respectivas funções densidade, domínios de falha e de segurança [14].	13
Figura 2.4 - Representação de $F_R(x)$ e de $f_S(x)$ num problema básico R-S [14].	14
Figura 2.5 - Representação da distribuição da margem de segurança Z [14].	15
Figura 2.6 - Transformação da função de estado limite no espaço das variáveis normais reduzidas [15].	18
Figura 2.7 - Transformação de uma variável não normal, X , numa variável normal reduzida, Y [15].	19
Figura 2.8 - Comportamento algorítmico [15].	21
Figura 2.9 - Algoritmo não converge devido à função de estado limite [15].	22
Figura 3.1 - Viga sujeita a flexão [15].	27
Figura 3.2 - Viga encastrada.	34
Figura 4.1 - Conceito de otimização robusta [12].	42
Figura 5.1 - Diagrama de execução do PROAES a partir da janela de comandos do MATLAB.	48
Figura 5.2 - Diagrama de execução do PROAES_FORM para executar uma análise de fiabilidade.	50
Figura 5.3 - Diagrama de execução para um problema de otimização determinístico.	51
Figura 5.4 - Diagrama de execução para um problema de otimização probabilístico.	52
Figura 6.1 - Pórtico, exemplo 1.	54
Figura 6.2 - Pórtico de 2 andares, exemplo 2.	57
Figura 6.3 - Resultados PROAES_FORM, exemplo 2.	60
Figura 6.4 - Sumário PROFILE, exemplo 2 (diferenças finitas).	61
Figura 6.5 - Sumário PROFILE, exemplo 2 (método contínuo).	61
Figura 6.6 - Configuração inicial do exemplo 3 [36].	65
Figura 6.7 - Convergência para o problema RBDO, exemplo 3.	68

Figura 6.8 - Configuração final da estrutura sobreposta à inicial, obtida pela formulação RBDO (exemplo 3).....	69
Figura 6.9 - Gráfico da variância de β_1 com as variáveis de projecto de 1 a 4, respectivamente de a) a d), (exemplo 3).....	71
Figura 6.10 - Frente de Parteo das soluções óptimas para a RDO, exemplo 3.	72
Figura 6.11 - Configuração final sobreposta à inicial, com a formulação RDO e $\alpha=0.8$ (exemplo 3).....	73
Figura 6.13 - Volume (a) e Variância (b) em função de α , exemplo 3.....	74
Figura 6.12 - Gráfico de β_1 (azul, +), β_2 (magenta, o), β_3 (preto, x), β_4 (vermelho, *) em função de α , exemplo 3.....	74
Figura 6.14 - Configuração do exemplo 4.....	76
Figura 6.15 - Representação das cargas aplicadas [56].....	77
Figura 6.16 - Convergência para o problema RBDO do exemplo 4.	82
Figura 6.17 - Frente de Parteo das soluções óptimas para a RDO, exemplo 4.	84

Índice de Tabelas

Tabela 3.1 - Comparação de resultados do exemplo.....	35
Tabela 5.1 - Ficheiros auxiliares criados pelo utilizador e pelo programa.....	46
Tabela 6.1 - Performances do exemplo 1.....	55
Tabela 6.2 - Variáveis de projecto do exemplo 1.....	55
Tabela 6.3 - Sensibilidades do exemplo 1 (calculadas pelo método contínuo).....	55
Tabela 6.4 - Sensibilidades do exemplo 1 (calculadas por diferenças finitas , $\Delta = 0,1\%$ do valor de cada variável).....	56
Tabela 6.5 - Performances do exemplo 2.....	58
Tabela 6.6 - Média e desvio padrão de cada tipo de variável, exemplo 2.....	58
Tabela 6.7 - Variáveis de projecto do exemplo 2.....	59
Tabela 6.8 - Comparação de resultados do exemplo 2.....	60
Tabela 6.9 - Derivadas de β em ordem aos valores médios das variáveis, calculadas no ponto X^* (exemplo 2).....	63
Tabela 6.10 - Derivadas de β em ordem aos desvios padrões das variáveis, calculadas no ponto X^* (exemplo 2).....	64
Tabela 6.11 - Grupos de elementos do exemplo 3.....	65
Tabela 6.12 - Modelo estocástico do exemplo 3.....	66
Tabela 6.13 – Performances do exemplo 3.....	66
Tabela 6.14 - LSF e índices de fiabilidade do exemplo 3, para configuração inicial.....	67
Tabela 6.15 - Variáveis de projecto para as formulações RBDO e RDO do exemplo 3....	68
Tabela 6.16 - Valores iniciais e finais para o problema RBDO, exemplo 3.....	69
Tabela 6.17 - Volume e Variância em função de α , exemplo 3.....	72
Tabela 6.18 - Valores optimizados do problema RDO com $\alpha = 0.8$, exemplo 3.....	73
Tabela 6.19 - Modelo estocástico do exemplo 4.....	78
Tabela 6.20 - Performances do exemplo 4.....	78
Tabela 6.21 - Performances do exemplo 4 (calculadas pelo programa ANSYS).....	78
Tabela 6.22 - Sensibilidades do exemplo 4 (calculadas pelo PROAES_3D).....	79
Tabela 6.23 - Sensibilidades do exemplo 4 (calculadas pelo método das diferenças finitas, com $\Delta = 0.1\%$).....	79
Tabela 6.24 - LSF e índices de fiabilidade do exemplo 4.....	80

Tabela 6.25 - Derivadas de β_1 em ordem aos valores médios das variáveis, exemplo 4. ..	80
Tabela 6.26 - Derivadas de β_1 em ordem aos valores desvios padrão das variáveis, exemplo 4.	80
Tabela 6.27 - Variáveis de projecto para as formulações RBDO e RDO do exemplo 4....	81
Tabela 6.28 - Valores iniciais e finais para o problema RBDO do exemplo 4.	82
Tabela 6.29 - Volume e Variância em função de α , exemplo 4.	84
Tabela 6.30 - Valores otimizados do problema RDO com $\alpha=0.8$, exemplo 4.	85

Siglas

FE	Elementos finitos (Finite Element)
FORM	Método de fiabilidade de primeira ordem (<i>First Order Reliability Method</i>)
FOSM	Método de fiabilidade do segundo momento (<i>First Order Second Moment</i>)
LSF	Função de estado limite (<i>Limit State Function</i>)
MPP	Ponto de falha mais provável (<i>Most Probable Point of failure</i>)
OR	Investigação operacional (<i>Operations Research</i>)
RD	Projecto robusto (<i>Robust Design</i>)
RBDO	Optimização com constrangimentos de fiabilidade (<i>Reliability-Based Design Optimization</i>)
RDO	Optimização robusta (<i>Robust Design Optimization</i>)
SORM	Método de fiabilidade de segunda ordem (<i>Second Order Reliability Method</i>)
SQP	Programação quadratic sequencial (<i>Sequential quadratic programming</i>)

Capítulo 1 - Introdução

1.1 Motivação e Enquadramento

Para projectar e fabricar produtos de elevada qualidade a menores custos são necessárias técnicas capazes de encontrar projectos que satisfaçam os requisitos e que melhor cumpram os objectivos pretendidos (função objectivo), especificados no início do projecto [1]. Desta forma, o engenheiro tem como tarefa escolher e encontrar os parâmetros que definam o projecto mais eficiente, obedecendo a restrições impostas. Este processo é conhecido como optimização.

Os primeiros trabalhos analíticos sobre optimização estrutural foram publicados por Maxwell [2] e Michell [3]. Mas a optimização estrutural moderna teve início após a década de 50, onde ocorreram desenvolvimentos significativos em três domínios importantes [4]:

- Os computadores digitais, que surgiram no início da década de 1950;
- Os métodos numéricos de *programação matemática*;
- Os métodos numéricos de análise de estruturas, como o *método dos elementos finitos* (FE).

Os computadores digitais deram um forte impulso à aplicação dos métodos numéricos de *Programação Linear* (SIMPLEX), então desenvolvidos pela comunidade de Investigação Operacional. Estes métodos foram aplicados para resolver problemas de optimização estrutural envolvendo estruturas reticuladas (treliças) [4].

Para além do avanço tecnológico ao nível da informática durante a década de 1950, sucederam-se avanços teóricos significativos na área dos métodos numéricos aplicados à mecânica estrutural. A descoberta do método dos elementos finitos (muitas vezes atribuída a Courant [5], enquanto Clough [6], criou o nome que designa o método) abriu pela primeira vez a possibilidade dos engenheiros projectistas realizarem a análise de estruturas verdadeiramente complexas [4].

Os desenvolvimentos acumulados até finais da década de 1950 encerravam um potencial enorme para a optimização estrutural, permitindo que esta conhecesse um franco progresso no período seguinte. O trabalho de Schmit [7], é considerado hoje em dia como o marco que separa os desenvolvimentos iniciais da optimização estrutural e a optimização estrutural moderna. Basicamente, este artigo adquiriu uma importância única no momento da sua publicação porque, por um lado, introduziu a ideia de combinar a análise estrutural por elementos finitos com os métodos numéricos de optimização e, por outro lado, demonstrou a viabilidade deste processo na resolução de problemas reais [4].

A optimização estrutural moderna, como é reconhecida desde 1960 até aos nossos dias, tem sido marcada por sucessivos progressos ao nível da teoria da optimização, dos métodos numéricos de optimização, do hardware e do software [4].

Este conceito, comparado com as metodologias até então utilizadas, contribuiu para o grande desenvolvimento na engenharia estrutural, uma vez que permitiu não só o aumento da complexidade das estruturas, do número de parâmetros de projecto e de constrangimentos, como também aumentou o potencial dos programas de elementos finitos [8]. No entanto, a optimização estrutural começou por utilizar parâmetros determinísticos. Verificou-se contudo que a aplicação destas técnicas conduzia a projectos óptimos muito sensíveis a variações inesperadas das condições de funcionamento. Por exemplo, se os valores dos parâmetros se afastarem dos preconizados, por deficiências na construção, ou se os carregamentos aplicados forem diferentes dos assumidos durante a optimização, uma estrutura optimizada poderá ter maior tendência para colapsar.

Com o objectivo de aproximar o mais possível o projecto óptimo da realidade, procurou incluir-se a incerteza na formulação dos problemas de optimização [8], [9].

Uma estrutura está normalmente exposta a incertezas induzidas por diversos factores, tais como, propriedades dos materiais, imprecisão geométrica e variação dos carregamentos, etc [10]. A fiabilidade estrutural procura modelar matematicamente essa incerteza através de variáveis aleatórias. Utiliza conceitos como a probabilidade de falha e o índice de fiabilidade.

Em geral, existem dois tipos de formulações de optimização estrutural que foram desenvolvidos para lidar com a incerteza dos parâmetros: optimização com constrangimentos de fiabilidade (*Reliability-based design optimization* - RBDO) e optimização robusta (*Robust Design Optimization* - RDO) [10].

A RBDO procura sistemas estruturais seguros e optimizados, assumindo a variabilidade dos parâmetros relacionados com as propriedades dos materiais, imprecisão geométrica e variação dos carregamentos. A RDO tenta encontrar esses sistemas que também têm variabilidade mínima de resposta estrutural [11]. Visa reduzir a variabilidade do desempenho estrutural causada por flutuações regulares e, melhorar a *performance* das estruturas [9], [12].

Uma vez que a RDO é uma temática recente, inovadora e ainda pouco desenvolvida, surgiu a motivação e o interesse de a investigar mais aprofundadamente, até porque esta oferece um grande potencial de investigação e ensino. Além disso, perspectiva-se que ganhe cada vez mais importância no âmbito do projecto estrutural.

1.2 Objectivos

A optimização estrutural procura minimizar uma função objectivo (por exemplo o peso da estrutura) considerando constrangimentos nas medidas de desempenho (por exemplo tensões e deslocamentos, que deverão ser inferiores a limites prescritos). Tendo-se constatado que as soluções óptimas obtidas tendiam a ser muito sensíveis a pequenas variações dos parâmetros que definem a estrutura, e sabendo que muitos desses parâmetros têm, no caso de estruturas reais, variações aleatórias, têm vindo a ganhar importância as formulações de optimização robusta - RDO. Nestes casos, procura-se minimizar simultaneamente o peso ou outra medida da eficiência da estrutura e, simultaneamente, minimizar a variabilidade da sua resposta (medida, por exemplo, através do desvio padrão das tensões ou deslocamentos) que ocorre como consequência da variabilidade dos parâmetros.

O primeiro objectivo da dissertação é o estudo e aplicação de novas formulações de RDO na resolução de problemas de optimização estrutural.

O segundo objectivo consiste no desenvolvimento de um conjunto de programas, em ambiente MATLAB, vocacionado para o ensino e investigação de RBDO e de RDO, que constitua uma plataforma adequada para realizar este estudo. Os programas utilizam elementos barra ou viga 2D e 3D para efectuar a modelação das estruturas. Calculam as derivadas pelo método contínuo para esses elementos, utilizam o algoritmo FORM (*First Order Reliability Method*) para calcular a probabilidade de falha e, utilizando a *optimtool* do MATLAB, permitem resolver problemas de RBDO e RDO de forma versátil e eficiente.

Este tema foi escolhido pelo interesse e potencial que a RDO apresenta para o projecto de estruturas e foi grandemente inspirado no trabalho desenvolvido por Kiureghian et al. [13] na área da fiabilidade estrutural, que originou o programa FERUM, também desenvolvido em ambiente MATLAB.

1.3 Estrutura da Dissertação

A presente dissertação está dividida em 7 capítulos e um conjunto de anexos. A sua organização segue as seguintes linhas de orientação:

- Capítulo 1 (Introdução): neste capítulo é feita uma apresentação dos pontos essenciais a abordar, motivação, objetivos e organização do trabalho;
- Capítulo 2 (Fiabilidade estrutural): é apresentada uma introdução ao tema. É feita uma abordagem do conceito de incerteza, do conceito de estado limite, dos métodos de análise de segurança estrutural (particular destaque para o algoritmo FORM) e uma descrição sobre a noção de variáveis aleatórias e distribuições probabilísticas. É exposto o conceito de fiabilidade estrutural e são definidos os conceitos de probabilidade de falha e índice de fiabilidade;
- Capítulo 3 (Sensibilidades): é composto inicialmente por uma definição da análise de sensibilidades. É abordado o método “contínuo”, as formulações utilizadas e o método das diferenças finitas;
- Capítulo 4 (Optimização Robusta): neste capítulo apresentam-se os conceitos de otimização estrutural, RBDO, RDO e respectivas formulações e definições;
- Capítulo 5 (Programa MATLAB): este capítulo é dedicado ao funcionamento dos programas desenvolvidos, e são apresentadas as funcionalidades de cada módulo individualmente e em conjunto. É também explicada a interação entre os programas e a utilização da *optimtool* do MATLAB assim como os diferentes modos de execução;
- Capítulo 6 (Exemplos): neste capítulo é feita a validação dos programas através da comparação com resultados obtidos por outros autores. São apresentados 4 exemplos e respectivos resultados;
- Capítulo 7 (Conclusão): neste capítulo são reunidas as principais conclusões.

Capítulo 2 - Fiabilidade estrutural

2.1 Introdução

Em engenharia, a fiabilidade de uma estrutura é caracterizada pela forma como responde aos carregamentos impostos, à magnitude ou tipo destes e pela sua resistência e rigidez [14]. A resposta aos carregamentos é considerada satisfatória se cumprir alguns requisitos (tais como segurança de colapso, limitação nas deformações, variação das cotas, entre outros). Os requisitos são associados a estados limite que, quando são violados, tornam o projecto não satisfatório.

O estudo da análise de fiabilidade estrutural centra-se no cálculo e na previsão da probabilidade de falha (a probabilidade de os estados limite serem violados durante o tempo de vida do projecto) [14].

2.1.1 Incerteza

Em projecto estrutural, algumas incertezas podem ser controladas pelo projectista (que pode minimizar as incertezas dimensionais, recorrendo a tolerâncias mais apertadas) enquanto que outras não podem ser controladas (tais como variações nos carregamentos, condições climáticas e propriedades do material). A existência de incerteza conduziu à inclusão de modelos probabilísticos na formulação de projectos. A Figura 2.1 mostra várias fontes de incerteza que podem condicionar o projecto estrutural.

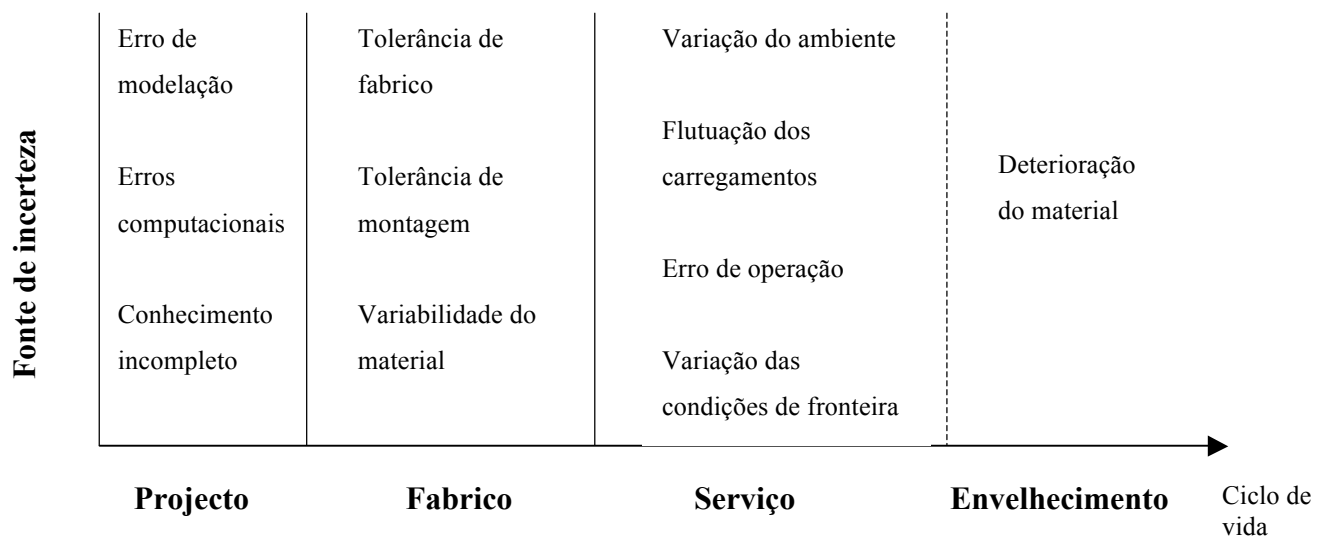


Figura 2.1 - Fontes de incerteza [9].

2.1.2 Variáveis aleatórias e distribuição probabilísticas

Os métodos da teoria da probabilidade tornam possível uma representação probabilística de um modelo estrutural, utilizando os conceitos de variável aleatória e função de distribuição [15]. Estas variáveis são associadas aos parâmetros que consideram a incerteza no modelo adoptado [16].

Uma variável aleatória é uma função que atribui um valor numérico a cada resultado individual de uma experiência aleatória, e tem como objectivo encontrar uma forma de passar da estrutura das probabilidades em espaços abstractos Ω para uma estrutura semelhante em \mathbb{R} .

Seja S o espaço amostral associado a uma experiência aleatória Ω . Uma função X , que associe a cada elemento $s \in S$ um número real $x(s)$, é denominada variável aleatória. Estas variáveis são então utilizadas para representar os parâmetros onde existe a incerteza [16].

As variáveis aleatórias podem ser classificadas de duas formas:

- **Variáveis aleatórias discretas:** matematicamente podemos dizer que as variáveis, cujo domínio é um conjunto finito ou infinito enumerável, são variáveis aleatórias discretas.
- **Variáveis aleatórias contínuas:** ao contrário das discretas, as variáveis aleatórias contínuas têm no domínio um conjunto infinito não enumerável.

Neste trabalho serão utilizadas vários tipos de distribuições probabilísticas, tais como: distribuição Normal, Log-normal, Gumbel e Rayleigh.

Nas secções seguintes a distribuição normal revela-se importante na resolução do algoritmo FORM e na transformação de Nataf.

2.1.3 Estados limite

Os requisitos de um projecto estão muitas vezes associados a estados limite. Recorrendo à análise feita para cada estado limite, podem-se identificar as variáveis aleatórias básicas que os influenciam. Por isso, os modelos do comportamento de uma estrutura não só devem definir cada estado, mas também a variáveis aleatórias básicas desses mesmos estados [15].

Através das variáveis aleatórias pode-se descrever um estado limite recorrendo à função de estado limite (*Limit State Function* - LSF):

$$Z = g(X) \tag{2.1}$$

onde X representa o conjunto das incertezas (variáveis básicas) do problema, g é a função estado limite, e Z a margem de segurança.

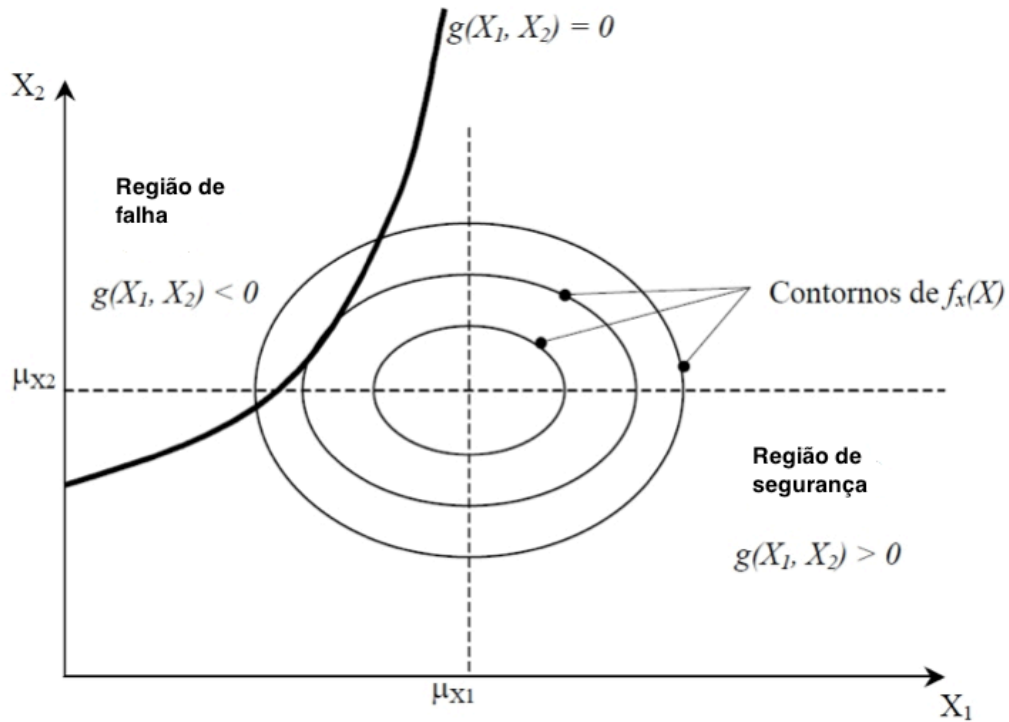


Figura 2.2 - Representação genérica da função $g(x)$ e da fronteira entre as regiões de segurança e falha para um problema com 2 variáveis aleatórias [15].

Verificando a Figura 2.2, é perceptível que, se a função de estado limite (LSF) for $Z = g(X)$, a condição $g(X) = 0$ representa a função de estado limite, ou seja, a fronteira entre a região de falha e a região de segurança [15]. Cada uma das zonas podem ser representadas da seguinte forma:

$$\begin{cases} Z = g(x) > 0 \Rightarrow \text{Zona de segurança} \\ Z = g(x) < 0 \Rightarrow \text{Zona de falha} \end{cases}$$

Em secções posteriores, esclarecer-se-á que a probabilidade de falha aparece representada por um integral de convolução correspondendo à totalidade dos casos em que a resistência é inferior a uma dada solicitação para toda a gama de solicitações possíveis.

2.2 Análise de Fiabilidade de estruturas

O problema básico de fiabilidade estrutural considera apenas a resistência, R , e a solicitação, S , cada uma descrita pela respectiva função densidade de probabilidade f_r e f_s . R e S poderão ser tensões, forças internas ou outras medidas do desempenho estrutural.

R - função das características geométricas e das propriedades dos materiais;

S - função das acções.

Caso S seja uma tensão, pode ser obtida a partir da carga aplicada Q através de uma análise estrutural (determinista ou aleatória).

Por conveniência, apenas a segurança de um elemento estrutural será considerada, e considera-se a falha desse elemento estrutural quando a sua resistência, R , é menor do que a tensão S actuando nele [14].

Então, matematicamente pode-se formular a função de estado limite da seguinte forma:

$$g(R, S) = R - S \quad (2.2)$$

Por conseguinte, a superfície que separa a região de falha da região de segurança passa a ser:

$$g(R, S) = R - S = 0 \quad (2.3)$$

A probabilidade de falha do elemento estrutural pode ser descrita da seguinte forma:

$$P_f = P(g(R, S) \leq 0) = P(R - S \leq 0) \quad (2.4)$$

As funções densidade f_r e f_s estão representadas na Figura 2.3, juntamente com a função densidade conjunta f_{rs} . Os elementos infinitesimais Δr e Δs , representam a probabilidade de R tomar um valor entre r e $r + \Delta r$ e de S tomar um valor entre s e $s + \Delta s$, respectivamente.

Na Figura 2.3 as equações 2.4 e 2.5 são representadas como domínio de falha, D, de modo que a probabilidade de falha torna-se:

$$P_f = P(R - S \leq 0) = \iint_D f_{rs}(r, s) dr ds \quad (2.5)$$

O domínio de falha, é: $D = \{R, S : G(R, S) \leq 0\}$.

Quando R e S são independentes, $f_{rs}(r, s) = f_r(r) f_s(s)$ e a probabilidade de falha pode-se escrever da seguinte forma:

$$P_f = P(R - S \leq 0) = \int_{-\infty}^{+\infty} \int_{-\infty}^{s \geq r} f_r(r) f_s(s) dr ds \quad (2.6)$$

Para qualquer variável aleatória X, a função de distribuição acumulada é determinada por:

$$F_X(x) = (P(X) \leq x) = \int_{-\infty}^x f_X(y) dy \quad (2.7)$$

Sendo $x \geq y$, segue-se que no caso especial quando R e S são independentes, a probabilidade de falha (2.6) pode ser escrita num integral único:

$$P_f = P(R - S \leq 0) = \int_{-\infty}^{+\infty} F_R(x) f_s(x) dx \quad (2.8)$$

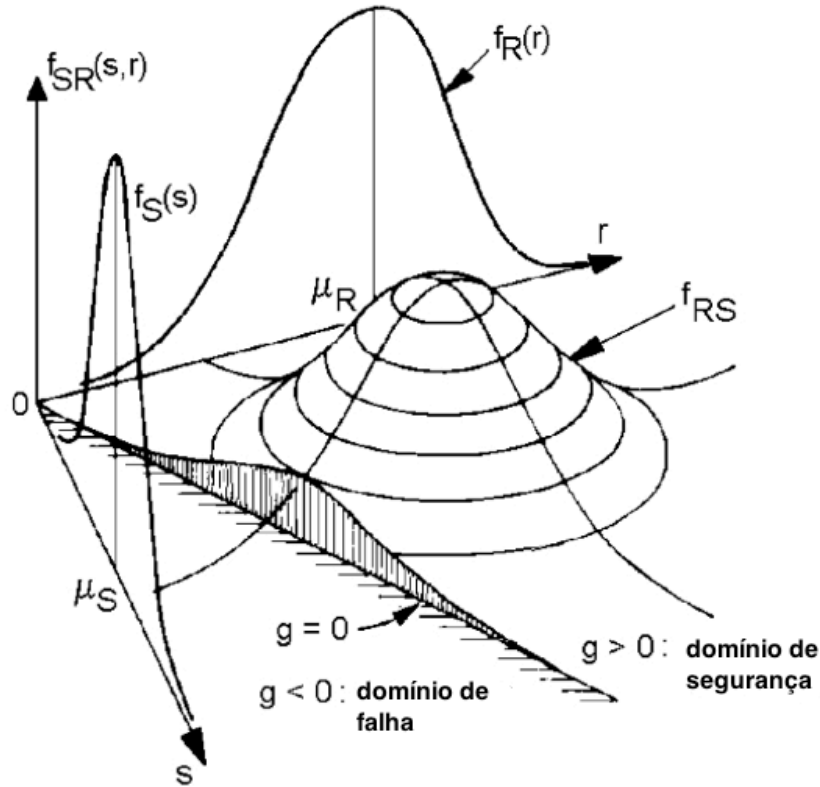


Figura 2.3 - Representação das variáveis aleatórias, respectivas funções densidade, domínios de falha e de segurança [14].

Este integral é também conhecido por integral de convolução, correspondendo à soma de todos os casos de solicitação para os quais a resistência não excede as acções [15], representado na Figura 2.4. $F_R(x)$ é a probabilidade de $R < x$, ou seja, a probabilidade de a resistência R do elemento ser menor que um valor x . O termo $f_s(x)$ representa a probabilidade de o efeito carga que está aplicado no componente ter um valor entre x e $x + \Delta x$ no limite em que $dx \rightarrow 0$.

Ao considerar todos os valores possíveis de x , ou seja, aplicando o integral a todo o x , a probabilidade total de falha é obtida, como pode ser visto na Figura 2.4, onde as funções de densidade f_R e f_S foram representadas ao longo do mesmo eixo, x [14].

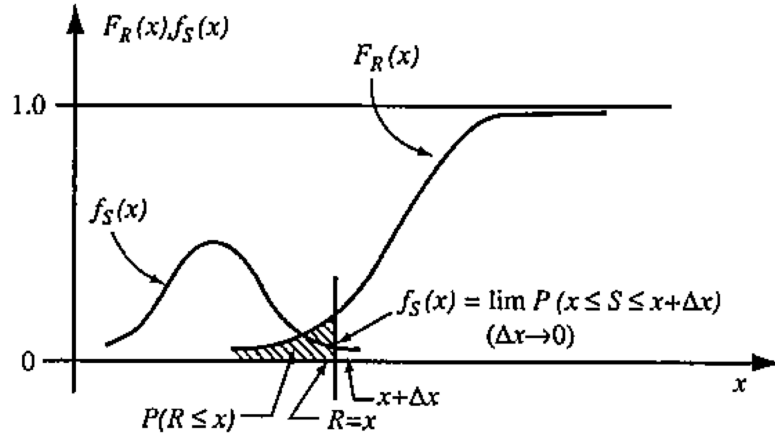


Figura 2.4 - Representação de $F_R(x)$ e de $f_S(x)$ num problema básico R-S [14].

Através de integração de $f_R(x)$ em (2.6), a integração foi reduzida. Esta operação é conveniente e útil, mas não é geral, e só é possível porque foi assumido que R é independente de S. Em geral, a dependência entre as variáveis deve ser considerada [14]. Uma alternativa para a expressão (2.8) é a seguinte:

$$P_f = \int_{-\infty}^{+\infty} [1 - F_S(x)] f_R(x) dx \quad (2.9)$$

Que é simplesmente a soma das probabilidades de falha de todos os casos em que a carga aplicada excede a resistência do elemento.

Como o integral de convolução (2.8) é bastante complexo apenas é possível a sua resolução analítica para um número limitado de problemas. Considerando as variáveis R e S como sendo variáveis aleatórias independentes com distribuição normal de média μ_R e μ_S , e variância σ_R^2 e σ_S^2 , respectivamente, o integral torna-se determinável [14].

A margem de segurança, $Z = R - S$, é neste caso uma variável aleatória com distribuição normal em que a média e variância se obtém por regras conhecidas de adição (subtração) de variáveis aleatórias com distribuição normal, então:

$$\left\{ \begin{array}{l} \mu_Z = \mu_R - \mu_S \\ \sigma_Z^2 = \sigma_R^2 + \sigma_S^2 \end{array} \right. \quad (2.10)$$

$$\left\{ \begin{array}{l} \sigma_Z^2 = \sigma_R^2 + \sigma_S^2 \end{array} \right. \quad (2.11)$$

Portanto a equação (2.5) pode ser escrita da seguinte forma:

$$P_f = P(R - S \leq 0) = P(g(R, S) \leq 0) = \Phi\left(\frac{0 - \mu_g}{\sigma_g}\right) = \Phi(-\beta) \quad (2.12)$$

onde Φ é a função distribuição da distribuição normal reduzida e β é o índice de fiabilidade, que é dado por:

$$\beta = \frac{\mu_g}{\sigma_g} \quad (2.13)$$

Na Figura 2.5 está representada a margem de segurança $Z = R - S$ e a região de falha associada a P_f (probabilidade de falha).

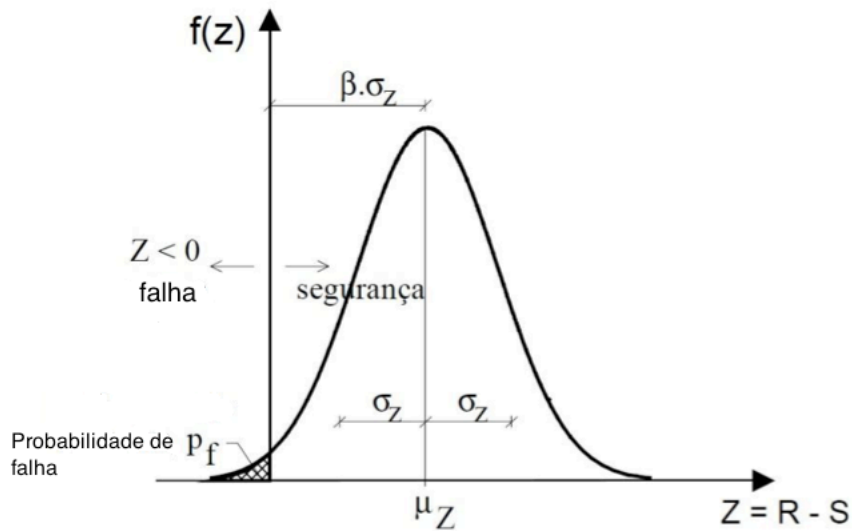


Figura 2.5 - Representação da distribuição da margem de segurança Z [14].

2.3 Métodos de Integração e Simulação

Tendo em conta que a integração analítica do integral de convolução (2.8) só é possível para alguns casos muito especiais e de interesse prático limitado, foi necessário desenvolver metodologias que permitissem a obtenção de P_f e de β em casos práticos e próximos da realidade. Destacam-se os seguintes trabalhos de investigação e metodologias: Cornell [17], Freudenthal et al. [18], método de fiabilidade estrutural FOSM (*First Order Second Method*), Hasofer et al. [19], método FORM (*First Order Reliability Method*), Rackwitz e Flessler [20], Chen e Lind [21], Wu e Wirsching [22], Liu [23], Ditlevsen e Madsen [24], método SORM (*Second Order Reliability Method*), e Rubinstein e Kroese [25], método de Monte Carlo.

Apesar das limitações, o método desenvolvido por Cornell [17], que introduz o conceito de índice de fiabilidade β , permitiu a quantificação indirecta da probabilidade de falha P_f e a avaliação da segurança de uma dada estrutura. A utilização desse índice no desenvolvimento da teoria da fiabilidade assume grande desenvolvimento particularmente nas novas metodologias de avaliação de segurança estrutural [15].

Essas novas metodologias, incluíram técnicas de probabilidade e estatística em modelos de elementos finitos e tinham como foco estruturas com comportamento linear ou não linear. Vanmarcke [26], Kiureghian [27], Spanos [28], são autores pioneiros, na década de noventa, desenvolveram aplicações e propostas sobre o estudo da fiabilidade, mais tarde aplicadas a comportamentos não lineares de estruturas, por Liu [23], Teigen [29], Rajashekhar [30], Zhang [31].

Dado a complexidade de resolver o integral de convolução (2.8) já mencionada, recorre-se a simplificações e/ou a métodos numéricos.

Os métodos de primeira ordem (FORM), que se baseam em aproximações lineares e os métodos de segunda ordem (SORM), baseados em aproximações quadráticas, são utilizados para a obtenção de soluções aproximadas à superfície de estado limite no ponto

mais provável de falha. O Método Monte Carlo efectua aproximações numéricas através de simulações.

Será analisado com mais pormenor o Algoritmo FORM, pois é utilizado nos programas desenvolvidos.

2.3.1 Algoritmo FORM

O método FORM tem como objectivo avaliar a fiabilidade de um sistema. Este método baseia-se em aproximações lineares da superfície de estado limite no ponto mais provável de rotura (MPP - *Most Probable Point of failure*), e foi inicialmente proposto por Hasofer e Lind [19] e mais tarde desenvolvido por Rackwitz e Fiessler [20].

O problema típico que o algoritmo FORM resolve aproximadamente, pode ser descrito pela equação seguinte:

$$P_f = P[g(X_i) \leq 0] \quad (2.14)$$

onde $g(X_i)$ representa a função de estado limite que depende das variáveis aleatórias X_i , que definem o problema de fiabilidade. As variáveis aleatórias X_i , normalmente, estão relacionadas com a resistência do sistema e cargas aplicadas.

$g(X_i) = 0 \rightarrow$ Corresponde à condição de estado limite;

$g(X_i) < 0 \rightarrow$ Define o domínio de falha, onde os critérios de segurança são violados;

$g(X_i) > 0 \rightarrow$ Define o domínio seguro.

A probabilidade de falha, P_f , pode ser calculada da seguinte forma:

$$P_f = \int \dots \int_{g(x) \leq 0} f_X(x) dx \quad (2.15)$$

onde $f_X(x)$ é a função densidade de probabilidade conjunta para as variáveis básicas X_i .

Quando essas variáveis não são normalmente distribuídas e são dependentes, é necessário transformar as variáveis básicas num conjunto de variáveis normais reduzidas, Y_i . O cálculo da probabilidade de falha resulta no problema de minimização que se segue:

$$\beta = \min(\sum_{i=1}^n Y_i^2)^{\frac{1}{2}} \quad (2.16)$$

$$\text{Sujeito a : } g(Y_i) = 0$$

Isto corresponde a encontrar o ponto de falha (ou de projecto) mais provável, y^* , que é o ponto sobre a superfície de falha mais próximo da origem do espaço normal padrão. O índice de fiabilidade β é a distância entre y^* e a origem do espaço normal padrão, representado na Figura 2.6.

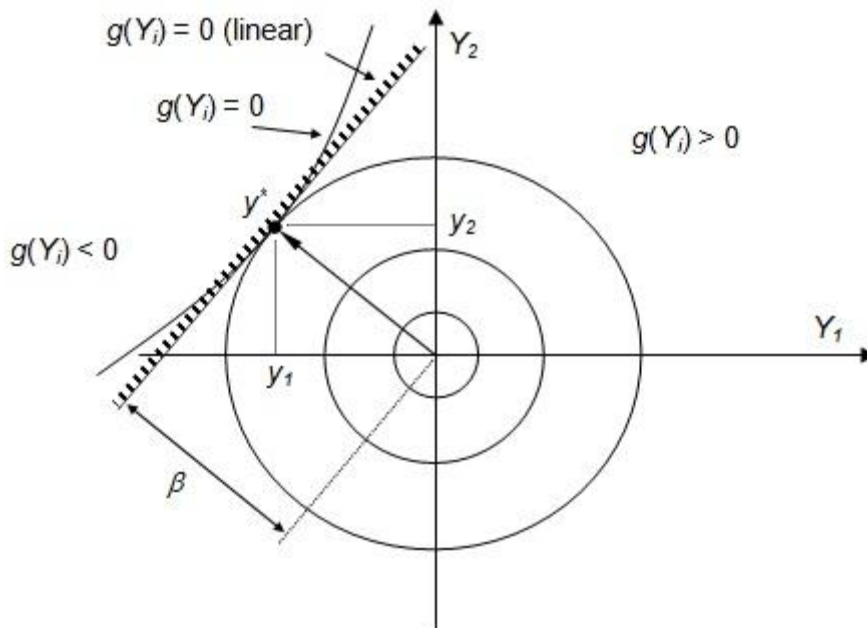


Figura 2.6 - Transformação da função de estado limite no espaço das variáveis normais reduzidas [15].

O mapeamento entre variáveis X e Y pode ser realizado através das transformações Rosenblatt ou Nataf. Estas transformações são baseadas no conceito de que qualquer ponto x no espaço original X é transformado em y no espaço normal padrão Y , Figura 2.7, tal que as distribuições cumulativas em x e y são iguais.

$$F_x(x) = \Phi(y) \quad (2.17)$$

Da equação (2.17) verifica-se que:

$$y = \Phi^{-1}[F_x(x)] \quad (2.18)$$

$$x = F_x^{-1}[\Phi(y)]$$

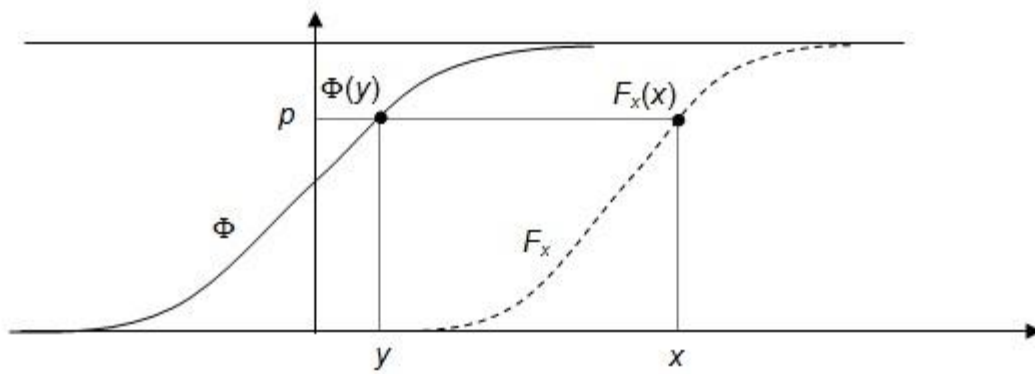


Figura 2.7 - Transformação de uma variável não normal, X , numa variável normal reduzida, Y [15].

O problema de minimização em (2.16), exige que os gradientes de g sejam calculados em ordem a Y_i . Isto pode ser obtido derivando a função estado limite em ordem a X_j e aplicando a regra da cadeia:

$$\frac{\partial g}{\partial X_j} = \frac{\delta g}{\delta Y_i} \frac{\delta Y_i}{\delta X_j} \quad (2.19)$$

Invertendo a relação, tem-se que:

$$\frac{\delta g}{\delta Y_i} = \frac{\delta g}{\delta X_j} \frac{\delta X_j}{\delta Y_i} \quad (2.20)$$

Na equação (2.20), o Jacobiano pode ser calculado a partir da seguinte transformação:

$$\frac{\delta X_j}{\delta Y_i} = \frac{\delta \Phi(y)}{\delta Y_i} \frac{\delta X_j}{\delta F_X(x)} = \frac{\Phi(Y_i)}{f(X_j)} \quad (2.21)$$

Na equação (2.21), $\Phi(Y_i)$ e $f(X_j)$ são respectivamente, a função densidade da distribuição normal padrão e a função de densidade para a variável X_j .

O algoritmo usado para resolver o problema de minimização (2.16) prossegue do seguinte modo.

É escolhido um ponto de partida $X_i^{(1)}$ no universo das variáveis básicas, por exemplo, considerando cada variável x_i igual ao seu valor médio. Então m é igual a 1 e o ciclo seguinte é executado até que o valor de β seja obtido com a tolerância desejada.

1. A função estado limite $g(X_i^{(m)})$ e o seu gradiente, $\frac{\delta g}{\delta X_i^{(m)}}$, são calculados;
2. As respectivas variáveis normais básicas $Y_i^{(m)}$ e o Jacobiano $\frac{\delta X_j^{(m)}}{\delta Y_i^{(m)}}$ são obtidos;
3. O gradiente da função estado limite no espaço normal padrão, $\frac{\delta g}{\delta Y_i^{(m)}}$ e $\alpha^{(m)}$ são calculados:

$$\alpha^{(m)} = \frac{\frac{\delta g}{\delta Y_i^{(m)}}}{\left| \frac{\delta g}{\delta Y_i^{(m)}} \right|} \quad (2.22)$$

4. A nova estimativa de Y , $Y_i^{(m+1)}$, é obtida a partir de:

$$y^{(m+1)} = -\alpha^{(m)} \cdot \beta^{(m+1)} \quad (2.23)$$

tal que:

$$\beta^{(m+1)} = \frac{g(Y_i^{(m)}) - \frac{\delta g}{\delta Y_i^{(m)}} \cdot y^{(m)}}{\left| \frac{\delta g}{\delta Y_i^{(m)}} \right|} = \frac{g(Y_i^{(m)})}{\left| \frac{\delta g}{\delta Y_i^{(m)}} \right|} + \beta^{(m)} \quad (2.24)$$

5. O ponto correspondente, $X_i^{(m+1)}$, é calculado no universo das variáveis básicas;
6. Depois de comparar $\beta^{(m)}$ e $\beta^{(m+1)}$ é tomada uma decisão sobre uma nova iteração.
Se for necessário, repetem-se todas as etapas atrás descritas.

Este algoritmo pode facilmente falhar se a função de estado limite não for linear (e até funções lineares passam a ser não lineares no espaço normal padrão). O comportamento algorítmico é mostrado na Figura 2.8:

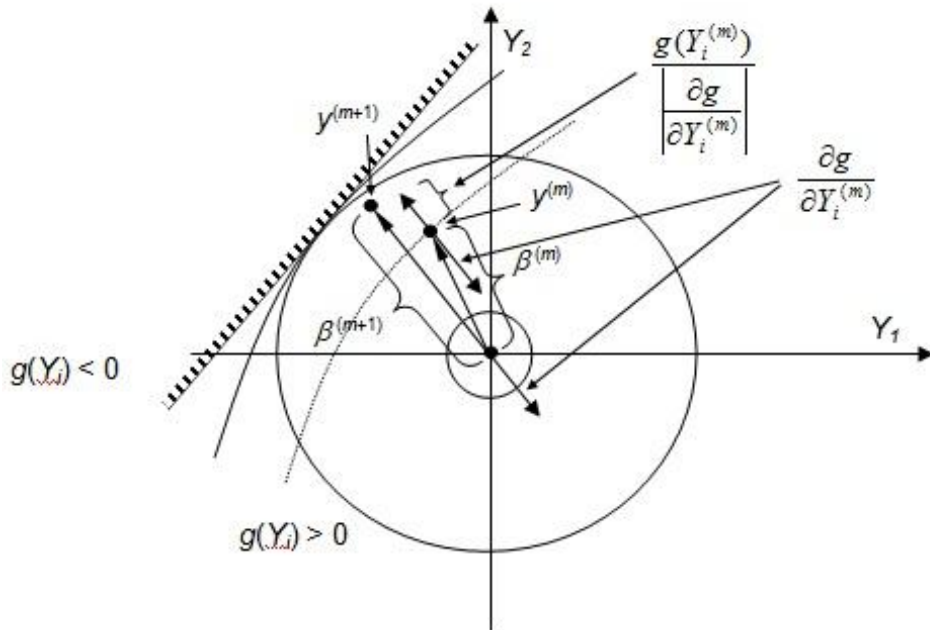


Figura 2.8 - Comportamento algorítmico [15].

Se a função limite de estado for extremamente não linear e convexa para a origem, como é representado na Figura 2.9, o algoritmo pode oscilar sem convergir.

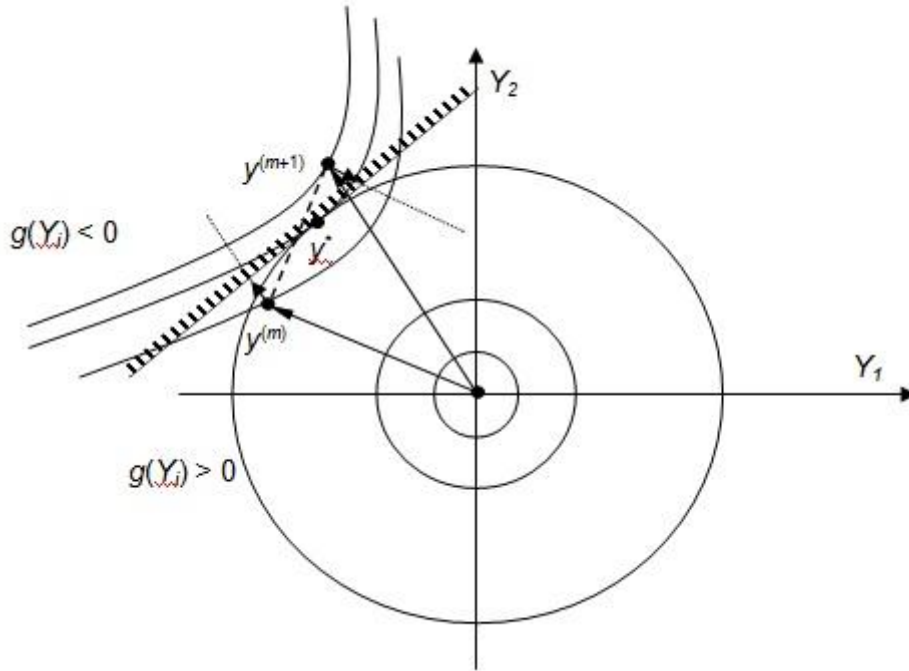


Figura 2.9 - Algoritmo não converge devido à função de estado limite [15].

Este algoritmo é original de Rackwitz-Fiessler (RF). Foi feita uma intensa pesquisa no sentido de melhorar este algoritmo. Ver, por exemplo [32].

A fim de melhorar a convergencia, o algoritmo Abdo-Rackwitz (HLRF) sugere uma pesquisa ao longo do trajecto que vai de $y^{(m)}$ para $y^{(m+1)}$ usando $z \in [0,1]$ e uma função de interpolação quadrática [33].

$$y(z) = (1 - z) \cdot y^{(m)} + z \cdot y^{(m+1)} \quad (2.25)$$

Zhang e Der Kiureghin [34], também propuseram um algoritmo (iHLRF) para realizar uma pesquisa para o ponto z que minimiza a função de mérito $m(y)$:

$$m(z) = \frac{\|y(z)\|^2}{2} + \lambda \cdot |g(z)| \quad (2.26)$$

em que:

$$\lambda > \frac{\|y(z)\|}{\left\| \frac{\delta g(z)}{\delta y} \right\|} \quad (2.27)$$

A equação (2.25) representa a função de Lagrange correspondente ao problema de optimização, e reflecte o facto de que o problema a ser resolvido é a minimização de y , com uma restrição em $g(z) = 0$, [35].

Nesta dissertação o método FORM é utilizado para avaliar a fiabilidade das estruturas calculando o índice de fiabilidade associado a cada LSF.

O programa MATLAB desenvolvido para este efeito baseia-se no trabalho realizado por Batista [15] e aplica o algoritmo iHLRF proposto por Zhang e Der Kiureghin [34].

Capítulo 3 - Análise de Sensibilidades

3.1 Introdução

A análise de sensibilidades tem como objectivo o cálculo dos gradientes das performances utilizadas para a avaliação de um projecto, relativamente às variáveis que o condicionam [36].

Nesta dissertação utiliza-se a análise de sensibilidades com o objectivo de determinar os gradientes da função de estado limite. Diversas metodologias para a determinação eficiente de campos de sensibilidade em problemas lineares e não-lineares estão bem estabelecidas desde a década de 80, destacando-se os trabalhos de Choi e Haug [37], Haug et al. [38] e Mota Soares [39]

A análise de sensibilidades visa calcular os gradientes, $\partial\psi / \partial X_i^{(m)}$, da resposta ou desempenho estruturais, ψ , utilizados na definição das funções de estado limite, $g(X_i^{(m)})$.

A principal vantagem do método contínuo, que é empregue nesta dissertação, é poupar significativamente o esforço computacional quando comparado com o método das diferenças finitas [15].

Os vários métodos de análise de sensibilidades foram inicialmente desenvolvidos para aplicação em optimização estrutural, em combinação com os métodos de programação matemática. Neste caso, os desempenhos estruturais estão relacionados com a função objectivo e os constrangimentos, enquanto os parâmetros estão relacionados com as variáveis do problema.

Existem duas abordagens para a análise de sensibilidade de projecto: sensibilidades discretas e sensibilidades contínuas [38]. Neste capítulo será apresentada uma formulação para o cálculo de sensibilidades, proposta por Cardoso [36], utilizando o método contínuo.

No caso das sensibilidades discretas o modelo numérico é diferenciado de modo a obter a sensibilidade requerida, enquanto que no caso das sensibilidades contínuas a equação diferencial subjacente é diferenciada e as equações resultantes são discretizadas usando o mesmo tipo de malha já utilizado na análise de elementos finitos ou uma outra malha criada especificamente para o efeito [38].

3.2 Análise de Sensibilidades – método contínuo

Em contraste com a equação matricial desenvolvida para as sensibilidades discretas, as equações das sensibilidades contínuas usam deslocamentos contínuos que satisfazem as condições fronteira. Embora as duas abordagens estejam relacionadas existem vantagens e desvantagens.

Comparando com sensibilidades discretas, a principal desvantagem das sensibilidades contínuas é a complexidade teórica.

As duas principais vantagens são:

- a) Uma rigorosa teoria matemática, sem incerteza associada com o erro de aproximação de dimensão finita;
- b) uma relação explícita de sensibilidades de projecto em termos de grandezas físicas, em vez de somatórios de derivadas de elementos matriciais.

Considerando a forma geral das equações de equilíbrio estático, obtidas a partir do princípio de trabalho virtual (ou a partir do princípio da energia potencial total mínima), sabe-se que:

$$a_x(z, \bar{z}) = l_x(\bar{z}) \quad (3.1)$$

Onde $a_x(z, \bar{z})$ é a forma bilinear relacionada com o trabalho virtual produzido por forças internas (ou variação da energia de deformação), $l_x(\bar{z})$ é a forma linear relacionada com o trabalho virtual produzido por forças externas (ou variação da energia potencial das

forças externas). De notar que agora z e \bar{z} representam, respectivamente, o deslocamento contínuo e o deslocamento virtual.

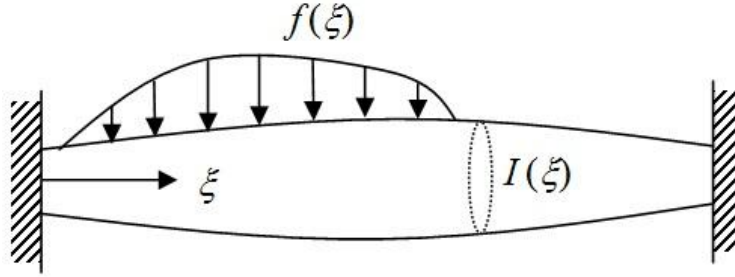


Figura 3.1 - Viga sujeita a flexão [15].

Para uma viga sujeita a flexão, como representado na Figura 3.1, com uma carga aplicada $f(\xi)$ e uma inércia variável $I(\xi)$, tem-se:

$$a_x(z, \bar{z}) = \int_0^L EI z_{\xi\xi\xi} \bar{z}_{\xi\xi\xi} d\xi \quad (3.2)$$

$$l_x(\bar{z}) = \int_0^L f z d\xi \quad (3.3)$$

Para obter as sensibilidades contínuas a partir da equação (3.1), é útil ter em conta o conceito de derivadas direccionais. Considere-se um projecto nominal designado por x e uma vizinhança de projecto descrita por uma variação arbitrária, δ_x , e um parâmetro τ .

$$x_\tau = x + \tau \delta_x \quad (3.4)$$

Supõe-se que ambos os termos em (3.1) são diferenciáveis em ordem a x . Assume-se também que a solução, z , das equações de estado, é diferenciável em ordem a x . A seguinte notação de derivada direccional é aplicada:

$$a'_{\delta_x}(z, \bar{z}) \equiv \frac{d}{d\tau} a_{x+\tau\delta_x}(\tilde{z}, \bar{z})|_{\tau=0} \quad (3.5)$$

Onde \tilde{z} indica o campo de deslocamentos de z com a dependência de τ suprimida, e \bar{z} não depende de τ . A notação principal representa a primeira variação com uma explícita dependência de x . Para o termo de carga linear tem-se:

$$l'_{\delta_x}(\bar{z}) \equiv \frac{d}{d\tau} [l_{x+\tau\delta_x}(\bar{z})] |_{\tau=0} \quad (3.6)$$

Para z a equação (3.7) representa a primeira variação da solução da equação (3.1) com x que varia na direcção de δ_x .

$$z' = z'(\xi; x, \delta_x) \equiv \frac{d}{d\tau} z(\xi; x + \tau \delta_x) |_{\tau=0} \quad (3.7)$$

Uma propriedade importante da variação do estado definido na equação (3.7) é o facto da ordem da variação em relação a x e da derivada parcial em ordem à variável independente ξ poder ser alterada. Isto significa que:

$$\begin{aligned} (z_i)' &= (z')_i \\ (z_{ij})' &= ((z_i)')_j = (z')_{ij} \end{aligned} \quad (3.8)$$

Esta característica é uma extensão directa da propriedade bem conhecida do cálculo das variações, em que a ordem da variação e das derivadas parciais pode ser alterada. Assume-se que as condições fronteira não dependem do projecto.

Considerando agora que a forma de energia bilinear da equação (3.1) é linear em z e que, dependendo do problema, envolve uma ou duas derivações em ordem a z . Assim, usando a regra da cadeia e as equações (3.5) e (3.7) obtém-se:

$$\frac{d}{d\tau} [a_{x+\tau\delta_x}(z(\xi; x + \tau \delta_x), \bar{z})] |_{\tau=0} = a'_{\delta_x}(z, \bar{z}) + a_x(z', \bar{z}) \quad (3.9)$$

Para qualquer campo de deslocamento virtual fixo, \bar{z} , e tendo variações dos dois lados da equação (3.1) e usando as equações (3.5) e (3.6) tem-se:

$$a_x(z', \bar{z}) = l'_{\delta_x}(\bar{z}) - a'_{\delta_x}(z, \bar{z}) \quad (3.10)$$

Considere-se agora uma medida da resposta estrutural que pode ser escrita na forma integral através da equação (3.11).

$$\psi = \int_{\Omega} g(z, \nabla z, x) d\Omega \quad (3.11)$$

Onde ∇z representa o gradiente de z e a função g é continuamente diferenciável. Para alguns problemas, o termo da equação (3.11), ∇z , pode ser substituído pelas segundas derivadas de z :

$$\psi' \equiv \frac{d}{d\tau} \left[\int_{\Omega} g(z(\xi; x + \tau \delta_x), \nabla z(\xi; x + \tau \delta_x), x + \tau \delta_x) d\Omega \right] \Big|_{\tau=0} \quad (3.12)$$

$$\psi' \equiv \int_{\Omega} [g_z z' + g_{\nabla z} \nabla z' + g_x \delta_x] d\Omega \quad (3.13)$$

Para obter uma expressão explícita de ψ' em ordem a δ_x , torna-se útil reescrever os dois primeiros termos da equação (3.13) em ordem a δ_x . Uma equação adjunta é introduzida, substituindo z' na equação (3.13) por um campo de deslocamento virtual, λ' , e igualando os termos que envolvam a forma de energia bilinear $a_x(\lambda, \bar{\lambda})$. A equação adjunta obtida para a variável adjunta é:

$$a_x(\lambda, \bar{\lambda}) \equiv \int_{\Omega} [g_z \lambda' + g_{\nabla x} \nabla \lambda] d\Omega \quad (3.14)$$

Uma vez que, $\bar{\lambda}$, é um campo de deslocamentos virtual, pode ser substituído por z' , ou seja:

$$a_x(\lambda, z') \equiv \int_{\Omega} [g_z z' + g_{\nabla z} \nabla z'] d\Omega \quad (3.15)$$

Da mesma forma, a equação (3.10) pode ser avaliada para \bar{z} igual a λ , ficando:

$$a_x(z', \lambda) = l'_{\delta x}(\lambda) - a'_{\delta x}(z, \lambda) \quad (3.16)$$

No entanto, a forma de energia bilinear é simétrica, como se pode verificar o lado esquerdo da equações (3.15) e (3.16) são iguais, obtendo-se assim o resultado desejado:

$$\int_{\Omega} [g_z z' + g_{\nabla z} \nabla z'] d\Omega = l'_{\delta x}(\lambda) + a'_{\delta x}(z, \lambda) \quad (3.17)$$

Substituindo este resultado na equação (3.13) é possível obter uma expressão para a sensibilidade da resposta, ψ , que pode ser calculada uma vez obtido o actual campo de deslocamentos z através da equação (3.1), e o campo de deslocamentos adjunto, λ , através da equação adjunta (3.14) [15].

$$\psi = \int_{\Omega} g_x \delta_x d\Omega + l'_{\delta x}(\lambda) + a'_{\delta x}(z, \lambda) \quad (3.18)$$

A equação (3.18) juntamente com as equações (3.1) e (3.14) podem agora ser discretizadas pelo método dos elementos finitos e, se possível, com a mesma malha. A sensibilidade de resposta pode ser calculada, respeitando os seguintes passos:

1. Resolver um sistema de equações lineares, resultantes da equação (3.1) e armazenar a matriz de rigidez na forma decomposta.

2. Calcular o vector de carga resultante da equação (3.14) que depende da resposta estrutural solicitada, e efectuar uma repetição da análise da estrutura com a mesma matriz de rigidez e considerando um vector de carga diferente para cada resposta.
3. Calcular a sensibilidade desejada fazendo o somatório dos três termos da equação (3.18) recorrendo a z e λ obtidos anteriormente. Repare-se que os últimos dois termos da equação (3.18) apenas dizem respeito ao produto interno entre o vector e a matriz, e somente aos termos originais do vector de carga e da matriz de rigidez que dependem do parâmetro x .

A fim de esclarecer o uso da equação (3.18), no cálculo da sensibilidade do deslocamento na extremidade da viga encastrada da Figura 3.2, em relação à variação da inércia da secção transversal, I e em relação à variação do módulo de elasticidade, E , seguem-se três passos:

1. A viga deve ser analisada e a solução de deslocamentos, z , para as cargas aplicadas deve ser armazenada.
2. Define-se uma equação adjunta com a ajuda da matriz de rigidez e um vector de carga adjunto. Esta carga adjunta é calculada a partir da equação (3.14) e depende da resposta estrutural. Para um deslocamento num nó ao longo de um grau de liberdade, a carga adjunta corresponde a uma carga unitária aplicada no nó e no grau de liberdade pretendido. Ao resolver a equação adjunta, uma variável adjunta é obtida e armazenada.
3. Calculam-se os três termos da equação (3.18). A resposta, g , e a carga aplicada, l , não dependem de I e, portanto, os dois primeiros termos de (3.18) são zero. O terceiro termo é obtido retirando a variação da equação (3.2) em relação a I , e a E , e obtendo respectivamente as equações (3.19) e (3.20).

$$a'_{\delta x}(z, \lambda) = \int_0^L E z_{\xi\xi} \lambda_{\xi\xi} d\xi \quad (3.19)$$

$$a'_{\delta x}(z, \lambda) = \int_0^L I z_{\xi\xi} \lambda_{\xi\xi} d\xi \quad (3.20)$$

Esta equação deve ser calculada para cada elemento finito da viga de modo a obter-se a sensibilidade pretendida.

A formulação utilizada para o cálculo de sensibilidades em relação a variações na posição no espaço ou variações de configuração foi proposta por Cardoso [36]. Representar-se-ão as equações finais, para elementos do tipo barra 2D e 3D e viga 2D e 3D, que foram incluídas no programa desenvolvido.

Para os elementos barra 2D e 3D apenas são consideradas as deformações sob a acção de cargas axiais, e a expressão é:

$$a'_V(z, \bar{z}) = - \int_0^L EA \left[z_{x,x} \bar{z}_{x,x} V_{x,x} - (z_{x,x} \bar{z}_{y,x} + z_{y,x} \bar{z}_{x,x}) V_{y,x} - (z_{x,x} \bar{z}_{z,x} + z_{z,x} \bar{z}_{x,x}) V_{z,x} \right] dx_x \quad (3.21)$$

Nesta equação, V representa a velocidade, isto é, a variação de posição no espaço.

No elemento viga 2D é considerada tracção/compressão e flexão em torno de z . A tracção/compressão é calculada pela equação (3.21) e a expressão total para o cálculo das sensibilidades do elemento viga 2D é:

$$a'_V(z, \bar{z}) = - \int_0^L EA \left[z_{x,x} \bar{z}_{x,x} V_{x,x} - (z_{x,x} \bar{z}_{y,x} + z_{y,x} \bar{z}_{x,x}) V_{y,x} - (z_{x,x} \bar{z}_{z,x} + z_{z,x} \bar{z}_{x,x}) V_{z,x} \right] dx_x - \int_0^L EI_z \left[3z_{y,xx} \bar{z}_{y,xx} V_{x,x} + (z_{y,x} \bar{z}_{y,xx} + z_{y,xx} \bar{z}_{y,x}) V_{x,xx} \right] dx_x \quad (3.22)$$

A expressão para o elemento viga 3D, considerando os efeitos de tracção/compressão, flexão em torno de y e z e da torção, pode ser escrita da seguinte forma:

$$\begin{aligned}
 a'_V(z, \bar{z}) = & - \int_0^L EA \left[z_{x,x} \bar{z}_{x,x} V_{x,x} - (z_{x,x} \bar{z}_{y,x} + z_{y,x} \bar{z}_{x,x}) V_{y,x} - (z_{x,x} \bar{z}_{z,x} + z_{z,x} \bar{z}_{x,x}) V_{z,x} \right] dx_x \\
 & - \int_0^L EI_y \left[3z_{z,xx} \bar{z}_{z,xx} V_{x,x} + (z_{z,x} \bar{z}_{z,xx} + z_{z,xx} \bar{z}_{z,x}) V_{x,xx} + 2(z_{z,yx} \bar{z}_{z,xx} + z_{z,xx} \bar{z}_{z,yx}) V_{y,x} + \right. \\
 & \quad \left. + (z_{z,y} \bar{z}_{z,xx} + z_{z,xx} \bar{z}_{z,y}) V_{y,xx} \right] dx_x \\
 & - \int_0^L EI_z \left[3z_{y,xx} \bar{z}_{y,xx} V_{x,x} + (z_{y,x} \bar{z}_{y,xx} + z_{y,xx} \bar{z}_{y,x}) V_{x,xx} - 2(z_{y,zx} \bar{z}_{y,xx} + z_{y,xx} \bar{z}_{y,zx}) V_{z,x} - \right. \\
 & \quad \left. - (z_{y,z} \bar{z}_{y,xx} + z_{y,xx} \bar{z}_{y,z}) V_{z,xx} \right] dx_x \\
 & - \int_0^L GJ \left[\phi_{x,x} \bar{\phi}_{x,x} V_{x,x} - (\phi_{x,x} \bar{\phi}_{y,x} + \phi_{y,x} \bar{\phi}_{x,x}) V_{y,x} - (\phi_{x,x} \bar{\phi}_{z,x} + \phi_{z,x} \bar{\phi}_{x,x}) V_{z,x} \right] dx_x
 \end{aligned} \tag{3.23}$$

3.3 Método das diferenças finitas

O método de diferenças finitas permite calcular aproximações de derivadas. A grande vantagem deste método é poder ser utilizado mesmo que a derivada analítica não seja conhecida. A forma mais simples de calcular os gradientes de uma função pelo método das diferenças finitas é usar a aproximação de primeira ordem, denominada diferença progressiva, dada pela seguinte equação.

Seja a função $f(x)$, onde x é a variável de projecto.

$$\frac{\Delta f}{\Delta x} = \frac{f(x + \Delta x) - f(x)}{\Delta x} \tag{3.24}$$

Outra aproximação por diferenças finitas bastante usada é a diferença finita central, que é dada pela equação seguinte.

$$\frac{\Delta \mu}{\Delta x} = \frac{\mu(x + \Delta x) - \mu(x - \Delta x)}{2 \Delta x} \tag{3.25}$$

Um dos problemas na utilização deste método é a escolha do tamanho da perturbação Δx , sendo que um valor muito pequeno leva a erros de arredondamento, causados pela forma como os números reais são representados nos computadores. Por outro lado, um valor elevado leva a erros de truncamento pois a derivada só é exacta quando Δx tende para zero. Perturbações relativas entre 10^{-4} a 10^{-8} geralmente levam a bons resultados, o suficiente para aplicações práticas.

O grande problema deste método é o custo computacional. No caso de serem usados modelos estruturais de elementos finitos, é necessário fazer uma nova análise completa para cada variável existente quando se usam diferenças progressivas ou duas análises quando se usam diferenças centrais. No entanto é de fácil implementação e por isso muito usado em optimização estrutural e em análise de fiabilidade [15].

3.4 Exemplo

Considere-se a viga encastrada representada na Figura 3.2, as sensibilidades do deslocamento da extremidade em relação à variação da inércia da secção transversal, I , e à variação do módulo de elasticidade E , são:

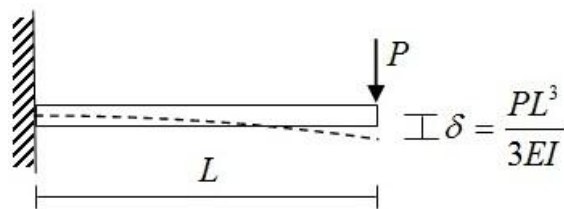


Figura 3.2 - Viga encastrada.

$$\delta = \frac{PL^3}{3EI} \Rightarrow \begin{aligned} \frac{d\delta}{dI} &= -\frac{PL^3}{3EI^2} \\ \frac{d\delta}{dE} &= -\frac{PL^3}{3IE^2} \end{aligned}$$

Neste caso, a resposta estrutural é o deslocamento na extremidade, $\psi = \delta$, a partir da qual pode ser gerada uma função de estado limite, $g(X)$, (por exemplo, $g(X) = \delta_{MAX} - \delta$). O vector X contém os parâmetros $x_1 = I$ e $x_2 = E$ e, uma aproximação válida da expressão que descreve a dependência de δ em I torna-se disponível [15].

Fornecendo alguns dados do problema, é possível calcular as derivadas (em ordem a I e E) usando os dois métodos (diferenças finitas e contínuo) e comparar os resultados com os valores obtidos a partir do cálculo analítico.

Dados do problema:

$$P = 100 N$$

$$I = 5,5498 \times 10^{-7} \text{ mm}^4$$

$$E = 200 \times 10^9 \text{ Pa}$$

$$L = 2 \text{ m}$$

Tabela 3.1 - Comparação de resultados do exemplo.

Método		$d\delta / dI$	$d\delta / dE$
Analítico		$4,32896 \times 10^3$	$1,20124 \times 10^{-14}$
Diferenças finitas	$\Delta x_1 = 10^{-3}$	$4,32464 \times 10^3$	$1,20004 \times 10^{-14}$
	$\Delta x_2 = 10^{-4}$	$4,32853 \times 10^3$	$1,20112 \times 10^{-14}$
	$\Delta x_3 = 10^{-5}$	$4,32892 \times 10^3$	$1,20123 \times 10^{-14}$
Contínuo		$4,32896 \times 10^3$	$1,20124 \times 10^{-14}$

Verifica-se que o método contínuo produz resultados, para o cálculo das derivadas, muito mais precisos do que o método de diferenças finitas. Os resultados determinados por diferenças finitas progressivas são apenas aproximados, no entanto verifica-se que a aproximação melhora quando se reduz Δx .

Comparando os resultados calculados pelo método contínuo com os resultados da solução analítica, pode-se afirmar que são esses resultados são exactos, para a precisão representada.

Nesta dissertação o método contínuo de análise de sensibilidades é empregue para calcular as derivadas das funções de estado limite utilizadas pelo método FORM e para calcular as derivadas da função objectivo e dos constrangimentos pelo algoritmo de optimização.

O programa MATLAB desenvolvido combina análise de sensibilidades e análise por elementos finitos e baseia-se no trabalho inicial de Batista [15], que foi extendido de modo a considerar elementos viga 3D, performances de tensão e variáveis de configuração.

Capítulo 4 - Optimização Robusta

4.1 Optimização Estrutural

A optimização estrutural é uma disciplina que engloba um conjunto de teorias e métodos que procuram obter a estrutura que desempenha mais eficientemente a função pretendida [4]. Ou seja, procura obter o conjunto de parâmetros que definem a estrutura com as características mais adequadas (por exemplo, com o peso mínimo), obedecendo a restrições impostas no projecto (como por exemplo, tensões, deslocamentos ou outras medidas do desempenho dentro da gama de valores admissíveis). A formulação clássica usada para um problema de optimização estrutural é:

$$\begin{aligned} & \text{Minimizar} \quad f(X) \\ & \text{Sujeito a:} \quad g_i(X) \leq 0, \quad i = 1, \dots, p \\ & \quad \quad \quad h_m(X) = 0, \quad m = 1, \dots, k \\ & \quad \quad \quad e \quad {}^lX_j \leq X_j \leq {}^uX_j, \quad j = 1, \dots, n \end{aligned} \tag{4.1}$$

Onde $X = (x_1, \dots, x_n)$, é o vector das n variáveis de projecto, $f(X)$ é a função objectivo e $g_i(X)$ e $h_m(X)$ são os p constrangimentos de desigualdade e os k constrangimentos de igualdade, respectivamente. Os limites inferior e superior de variação das variáveis de projecto são lX_j e uX_j , respectivamente.

4.2 Optimização com constrangimentos de fiabilidade - RBDO

O estudo da segurança de uma estrutura, considerando as várias incertezas existentes na fase de projecto, e que são apresentadas na Figura 2.1, pode ser feito usando as metodologias desenvolvidas no âmbito da análise de fiabilidade de estruturas. Nesse caso, a incerteza é modelada através de variáveis aleatórias caracterizadas por terem distribuições probabilísticas (distribuição normal, lognormal, etc...) e parâmetros associados (média, desvio padrão). A optimização de uma estrutura considerando essa incerteza, designa-se por optimização com constrangimentos de fiabilidade e utiliza normalmente a seguinte formulação:

$$\begin{aligned} & \text{Minimizar } f(\mu_x) \\ & \text{Sujeito a: } g_i(X) \leq 0, \\ & \quad e \quad {}^l\mu_{x_r} \leq \mu_{x_r} \leq {}^u\mu_{x_r}, \quad r = 1, \dots, n_x \end{aligned} \quad (4.2)$$

onde $f(\mu_x)$ é a função objectivo, normalmente o peso da estrutura, μ_x o vector das variáveis de projecto, cujos limites inferiores e superiores estão especificados nos vectores ${}^l\mu_{x_r}$ e ${}^u\mu_{x_r}$ respectivamente, e g_i o vector dos constrangimentos. No vector g_i está incluído pelo menos um constrangimento de fiabilidade e o vector das variáveis de projecto, μ_x , pode também incluir parâmetros das distribuições probabilísticas utilizadas na definição estatística das variáveis aleatórias. Consoante se avalie a fiabilidade da estrutura através da probabilidade de falha P_f ou do índice de fiabilidade β , assim os constrangimentos de fiabilidade poderão tomar uma das seguintes formas:

$$g_i = \frac{P_f - P_{\max}}{P_{\max}} \leq 0 \quad ; \quad g_i = \frac{\beta_{\min} - \beta}{\beta_{\min}} \leq 0 \quad (4.3)$$

onde P_{\max} é o limite superior imposto à probabilidade de falha associada a um dos possíveis modos de rotura da estrutura, definido por uma função de estado limite, e β_{\min} é o índice de fiabilidade mínimo correspondente.

Em geral, o projecto com base na fiabilidade é computacionalmente caro, pois exige muito mais avaliações das várias funções envolvidas do que um problema de optimização determinístico correspondente.

A abordagem mais utilizada para resolver este problema combina as técnicas FORM ou SORM com algoritmos de optimização baseados em gradientes. Tradicionalmente utiliza dois ciclos, um correspondente ao algoritmo de optimização e o outro ao algoritmo FORM [40]. Para cada iteração do algoritmo de optimização é necessário calcular a fiabilidade associada às várias funções de estado limite, o que requer várias iterações do algoritmo FORM, isto é, um ciclo está incorporado no outro. Contudo, outras abordagens têm sido utilizadas, por exemplo a utilização de algoritmos genéticos para realizar a optimização e o método Monte Carlo para calcular a fiabilidade [41]. Também Kaymaz et al. [42] utilizou métodos de superfície de resposta, substituindo o modelo de elementos finitos real e combinou esses métodos com simulação de Monte Carlo para ultrapassar a dificuldade de cálculo da fiabilidade em termos de custo computacional para os problemas de optimização. Esta técnica é especialmente eficaz quando funções de desempenho não-lineares estão envolvidas. Kleiber [8] estudou problemas de RBDO de estruturas de treliça, geometricamente não lineares, a fim de superar as dificuldades de convergência da abordagem totalmente automatizada para grandes estruturas lineares.

Ressalte-se que, para além dos custos de construção ou perda causada diretamente pela falha catastrófica, algumas métricas relacionadas com os custos globais das estruturas foram tomadas em consideração na RBDO. Wen [43] estudou a optimização de projecto de estruturas contra riscos múltiplos com base em considerações de minimização do custo do ciclo de vida útil, incluindo custos relativos à construção, manutenção, operação, reparação e custos consequentes das falhas consequência. Uma das soluções do custo total do ciclo de vida esperado, é utilizada no processo de optimização.

A RBDO apresenta sérias limitações relacionadas principalmente com a baixa eficiência computacional ou os problemas de convergência. Além disso, apenas num pequeno número de casos especializados a informação estatística completa sobre

parâmetros estruturais e cargas está disponível. Os pressupostos inadequados sobre a distribuição probabilística pode levar a erros substanciais na análise de fiabilidade. Neste sentido, RBDO só pode ser usada eficazmente se as informações sobre a incerteza aleatória estiverem disponíveis ou forem suficientes para permitir uma análise de fiabilidade eficaz [9].

4.3 Optimização Robusta - RDO

Em optimização robusta pretende-se minimizar uma medida da eficiência da estrutura (por exemplo minimizar o peso) e, simultaneamente, minimizar a variabilidade da resposta da estrutura (por exemplo o deslocamento ou a tensão máxima num ponto). Na optimização robusta a solução óptima é aquela que confere um mais elevado grau de robustez, isto é, aquela em que o projecto é relativamente invariante face às incertezas do projecto [44].

A optimização robusta tem vindo a ganhar importância devido ao facto de se verificar que as soluções óptimas obtidas pelos métodos tradicionais de optimização estrutural são muito sensíveis a pequenas variações dos parâmetros que definem a estrutura. Um dos inconvenientes desses métodos tradicionais é o facto de não incluírem a incerteza e a variabilidade dos parâmetros que condicionam o comportamento das estruturas. A incerteza é uma característica inerente à natureza, existe sempre, logo não pode ser evitada [9]. Para tratar dessa variabilidade alguns autores utilizam descrições probabilísticas enquanto outros recorrem a descrições não-probabilísticas [1].

Os modelos que usam descrições não-probabilística não fornecem informações sobre a frequência de ocorrência das combinações extremas das incertezas e, portanto, podem ser mais conservadores que os modelos de descrição probabilística em algumas circunstâncias [10]. Utilizando descrições não-probabilísticas, surgiram os modelos convexos desenvolvidos por Ben-Haim e Elishakoff [45] e o modelo conjunto de intervalos (caso especial do modelo convexo - *interval set model*), que foi abordado por

vários autores, tais como Jiang [46] e Impollonia e Muscolino [47]. Outros modelos deste tipo utilizam modelos de conjuntos difusos (*fuzzy set model*).

Nos modelos convexos somente é necessário o conhecimento dos limites dos parâmetros incertos, e esses limites podem ser obtidos com base num pequeno número de amostras [48], [49].

Contudo, grande parte das formulações existentes são baseadas na descrição probabilística de incertezas [10]. Sandgren e Cameron [50], empregaram um algoritmo genético para a procura de projectos robustos, incorporando os desvios padrão das respostas estruturais nos constrangimentos. Em 2005, Lagaros propôs um método não-dominante utilizando algoritmos CEA, (Cascade Evolutionary Algorithm), para a RDO em estruturas [51]. Posteriormente, Doltsinis e Kang aplicaram a RDO em estruturas e com deformações inelásticas [52]. Guest e Igusa, estudaram a optimização de topologia de estruturas com carga e posições dos nós incertas, descritas por funções de densidade de probabilidade conjunta [53]. Harzheim e Warnecke [54], realizaram RDO em casos práticos de engenharia utilizando modelos de superfície de resposta. Chen estudou a robustez e optimização topológica sob carregamentos aleatórios, com incerteza associada às propriedades dos materiais e incertezas geométricas [55].

As abordagens RDO que empregam descrições probabilísticas estão bem estabelecidas. Geralmente dependem de dados precisos da distribuição de probabilidade dos parâmetros aleatórios, embora em algumas aplicações de engenharia reais estes dados possam ser difíceis de obter devido à falta de amostras suficientes. Em tais circunstâncias, os projectistas têm de fazer certas suposições na construção de um modelo probabilístico [10].

Apesar deste inconveniente, optou-se nesta dissertação por aplicar modelos probabilísticos de optimização robusta, RDO, devido ao facto de constituírem uma extensão natural das formulações usadas em optimização com constrangimentos de fiabilidade, RBDO. A intensa actividade de investigação sobre RBDO que ocorreu nas últimas décadas produziu uma base sólida que pode ser utilizada como suporte para a RDO.

O conceito de optimização robusta pode ser ilustrado na Figura 4.1.

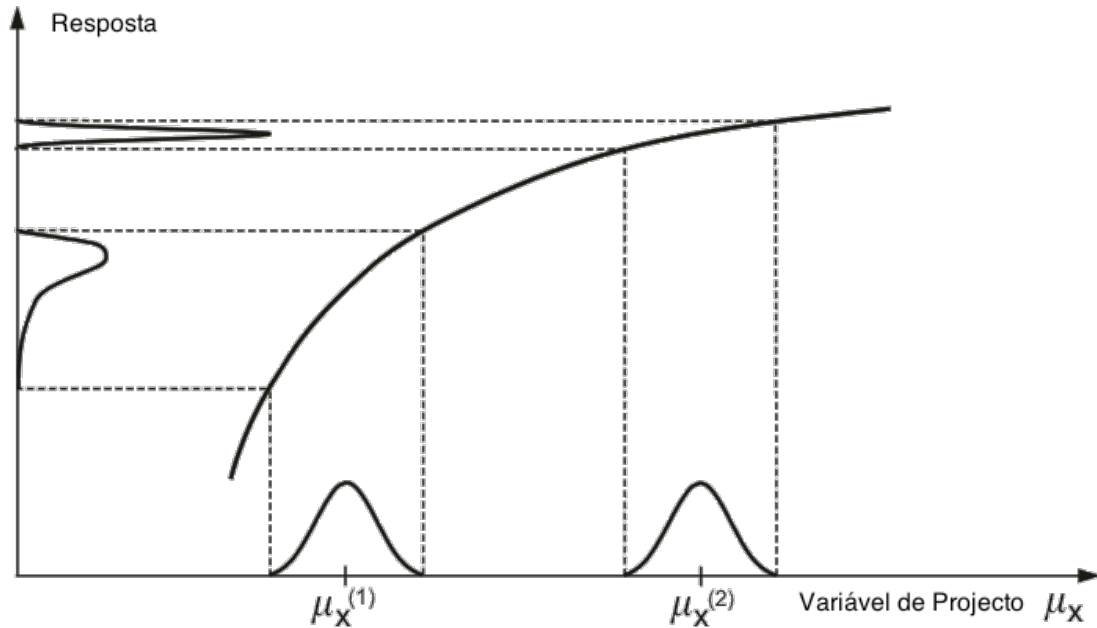


Figura 4.1 - Conceito de optimização robusta [12].

Assumindo uma variação constante da variável de projecto μ_X , que representa a média da variável aleatória X , a variabilidade da resposta é muito menor quando $\mu_X = \mu_{X(2)}$, do que quando $\mu_X = \mu_{X(1)}$, embora o valor médio da variável seja maior no primeiro caso. Em optimização robusta é desejável que se obtenha uma baixa variabilidade do desempenho, mesmo com custos de projecto superiores em detrimento de projectos caracterizados por valores mínimos da função objectivo (peso) mas com elevada sensibilidade à inevitável incerteza dos parâmetros, como é comum nas soluções obtidas pelos métodos de optimização estrutural clássicos [12]. Na Figura 4.1, a sensibilidade da resposta em relação à variável de projecto corresponde ao declive da função representada.

Ao contrário do que acontece na optimização estrutural clássica, não existe actualmente uma formulação única para os problemas de optimização robusta [1], uma das formulações usadas é [12]:

$$\begin{aligned}
\text{Min } f &= \frac{1-\alpha}{\hat{\mu}} \mu[f(d, X, P)] + \frac{\alpha}{\hat{\sigma}} \sigma[f(d, X, P)] \\
\text{Sujeito a: } &\mu[g_i(d, X, P)] - \beta_i \sigma[g_i(d, X, P)] \geq 0, i = 1, \dots, k_g \\
&\sigma[g_l(d, X, P)] \leq {}^u\sigma_l, l = 1, \dots, k_c \\
&{}^l d_j \leq d_j \leq {}^u d_j, j = 1, \dots, n_d \\
&{}^l \mu_{X_r} \leq \mu_{X_r} \leq {}^u \mu_{X_r}, r = 1, \dots, n_X
\end{aligned} \tag{4.4}$$

Onde $d \in R^{n_d}$ é o vector das variáveis determinísticas, $X \in R^{n_x}$ é o vector das variáveis aleatórias, $P \in R^{n_p}$ é o vector do parâmetros aleatórios e μ_X é o vector dos valores médios das variáveis X que varia durante a optimização. As funções $f(\cdot)$ e $g(\cdot)$ são, respectivamente, a função objectivo e as funções utilizadas para definir os constrangimentos. $\mu[\cdot]$ e $\sigma[\cdot]$ são respectivamente a média e o desvio padrão dessas funções, β é o índice de fiabilidade mínimo admissível, ${}^l d_j$ e ${}^u d_j$ são os limites mínimo e máximo de variação das variáveis determinísticas e ${}^l \mu_X$ e ${}^u \mu_X$ são os valores mínimos e máximos da média das variáveis aleatórias. O parâmetro $\alpha \in [0,1]$ permite seleccionar o peso relativo das duas parcelas que são usadas na definição da função objectivo. $\hat{\mu}$ e $\hat{\sigma}$ são valores utilizados para normalizar a função objectivo [12].

A optimização robusta permite alcançar projectos óptimos apesar da natureza aleatória dos parâmetros. A variabilidade da performance de um estrutura é tida em conta nestas formulações através do desvio padrão ou variância. Nesta dissertação desenvolveu-se um conjunto de programas capaz de efectuar RDO, recorrendo ao *optimtool* do MATLAB, e as formulações adoptadas são baseadas em Stocki [12].

Capítulo 5 - Programa PROAES

5.1 Introdução

O programa PROAES, é um programa que foi concebido em linguagem MATLAB e é composto por um conjunto de módulos que apresentam as seguintes capacidades:

1. Realizar a análise de modelos estruturais pelo método dos elementos finitos.
2. Calcular derivadas de funções associadas ao desempenho estrutural em ordem a variáveis de projecto.
3. Realizar análise de fiabilidade de modelos estruturais pelo método FORM.
4. Optimizar modelos de estruturas, considerando constrangimentos associados a medidas de desempenho estrutural e/ou medidas de fiabilidade.

Caso seja pretendida a análise de uma estrutura pelo método dos elementos finitos, apenas é utilizado o programa PROAES, no entanto se pretende calcular o índice de fiabilidade de uma estrutura, β , pelo método FORM, são usados os programas PROAES e PROAES_FORM. O programa PROAES calcula a resposta da estrutura e as suas derivadas, que servem para definir as funções de estado limite, enquanto o programa PROAES_FORM calcula os índices β associados a uma ou várias funções de estado limite que condicionam a segurança da estrutura. Na resolução de problemas de fiabilidade, as funções de estado limite podem estar associadas ao deslocamento num nó da estrutura e/ou à tensão admissível num ponto.

As funcionalidades de optimização requerem a utilização da ferramenta de optimização do MATLAB, *optimtool*.

5.1.1 Constituição dos programas

Nesta secção apresentam-se os ficheiros que resultaram do desenvolvimento e funcionamento do PROAES e PROAES_FORM.

Os ficheiros que constituem os programas são o *PROAES.m/PROAES_3D.m* e o *PROAES_FORM.m*, a cada um destes ficheiros estão associados ficheiros auxiliares e no caso do ficheiro PROAES_FORM estão também associadas subfunções. Os ficheiros auxiliares podem ser ficheiros de dados escritos pelo utilizador e ficheiros de resultados escritos pelo programa.

Tabela 5.1 - Ficheiros auxiliares criados pelo utilizador e pelo programa.

Ficheiros auxiliares	
Criados pelo utilizador	Devolvidos pelo programa
*.inp	*.out
*.frm	*.bpc

*.frm - ficheiro onde estão todos os dados sobre as variáveis envolvidas no problema em estudo, o modelo estocástico.

*.inp - ficheiro onde se podem encontrar todos os dados relativos ao modelo de elementos finitos usado no problema.

*.out - ficheiro onde são escritos os resultados.

*.bpc - ficheiro onde serão automaticamente registados os resultados calculados pelo PROAES_FORM, nomeadamente, a probabilidade de falha, o índice de fiabilidade, e o número de iterações que foram necessários para se atingir a convergência.

Caso seja pretendido realizar optimização recorrendo ao *optimtool*, é necessário que o utilizador crie duas funções, uma para definir a função objectivo, *myfunc.m* e outra para definir os constrangimentos, *mycon.m*.

5.2 Funcionamento do programa

Inicialmente o utilizador deve decidir o tipo de análise que pretende fazer, podem ser feitos 4 tipos de análises:

- Análise de elementos finitos e/ou sensibilidades;
- Análise de fiabilidade;
- RBDO;
- RDO.

5.2.1 Análise de elementos finitos e/ou sensibilidades

Existem 3 opções para executar o programa PROAES.

```
>> PROAES  
>> PROAES ('problema')  
>> PROAES ('problema', 'o')
```

Nos dois primeiros casos o programa é executado a partir da janela de comandos do MATLAB. No primeiro, o programa pergunta qual o nome do ficheiro *.inp ao utilizador através da interface gráfica e executa a análise de elementos finitos. No segundo não pergunta nada e executa a análise de elementos finitos, com o ficheiro de dados 'problema'.inp.

Caso seja executado com a última opção, o programa não pergunta nada, lê o ficheiro de dados '**problema**'.inp, lê em seguida o ficheiro **opt2ef**, executa a análise de elementos finitos e/ou de sensibilidades de acordo com o indicado em **opt2ef** e escreve os resultados em **ef2opt**. Esta possibilidade existe para que o programa possa ser usado para calcular funções de estado limite e suas derivadas, quando combinado com o programa PROAES_FORM, ou para calcular a função objectivo, os constrangimentos e suas derivadas quando usado em combinação com a ferramenta de optimização do MATLAB.

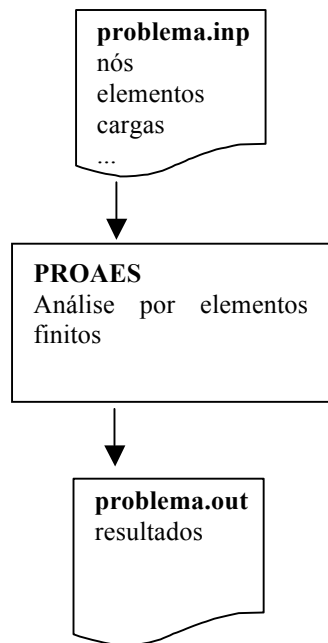


Figura 5.1 - Diagrama de execução do PROAES a partir da janela de comandos do MATLAB.

A Figura 5.1 mostra o diagrama de execução para as 2 primeiras opções, e a Figura 5.2 o diagrama correspondente à terceira opção, quando o PROAES é executado a partir do PROAES_FORM, ou quando se pretende gerar o ficheiro **ef2opt**.

O programa PROAES permite calcular derivadas de funções associadas ao desempenho estrutural em ordem a variáveis de projecto através do método contínuo de análise de sensibilidades. Essas derivadas podem ser usadas para calcular a fiabilidade de uma estrutura pelo método FORM ou para realizar a optimização de uma estrutura através de um algoritmo que necessite de derivadas.

A análise de sensibilidades pelo método contínuo permite obter valores muito precisos das derivadas numa fracção do tempo requerida pelo método das diferenças finitas, e não é sensível ao valor da perturbação no valor das variáveis, que conduz a erros de aproximação neste último método.

5.2.2 Análise de fiabilidade de estruturas pelo método FORM

O programa pode ser utilizado para calcular o índice de fiabilidade de uma estrutura, β , pelo método FORM. Nesse caso são usados os programas PROAES e PROAES_FORM. O programa PROAES calcula a resposta da estrutura e as suas derivadas, que servem para definir as funções de estado limite. O programa PROAES_FORM calcula os índices β associados a uma ou várias funções de estado limite que condicionam a segurança da estrutura.

O programa pode ser executado das seguintes formas:

```
>> PROAES_FORM  
>> PROAES_FORM('problema')  
>> PROAES_FORM('problema','o')
```

Nos dois primeiros casos o programa é executado a partir da janela de comandos do MATLAB. No primeiro, o programa pergunta qual o nome do ficheiro *.frm ao utilizador através da interface gráfica e executa a análise de fiabilidade. No segundo não pergunta nada e executa a análise de fiabilidade, com o ficheiro de dados '**problema**'.frm.

Caso seja executado com a última opção, o programa não pergunta nada, lê o ficheiro de dados '**problema**'.frm, lê em seguida o ficheiro **opt2form**, executa a análise fiabilidade de acordo com o indicado em **opt2form** e escreve os resultados em **form2opt**. Esta possibilidade existe para que o programa possa ser usado em combinação com a ferramenta de optimização do MATLAB para calcular a função objectivo, os constrangimentos e suas derivadas. Nesta última opção de execução, todo o conjunto de dados necessários para efectuar optimização, estão disponíveis.

No entanto para proceder a esta análise é necessário que o utilizador crie também o ficheiro, '**problema**'.inp que contém os dados do modelo de elementos finitos. Isto deve-se ao envolvimento entre o PROAES e o PROAES_FORM.

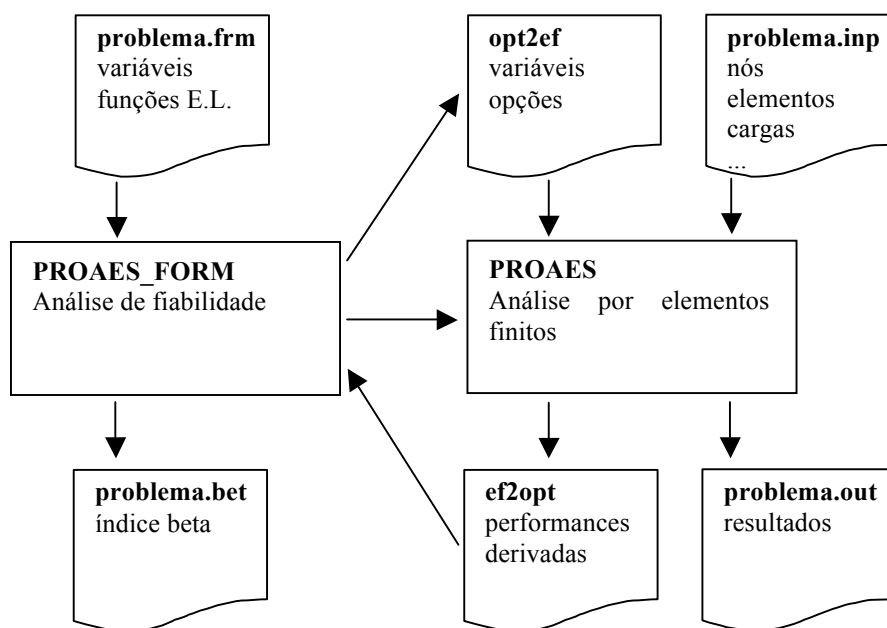


Figura 5.2 - Diagrama de execução do PROAES_FORM para executar uma análise de fiabilidade.

Na Figura 5.2 está representada a interação dos dois programas quando se executa uma análise de fiabilidade simples.

5.2.3 Optimização RBDO e RDO

Sabendo que calculam a resposta de uma estrutura e estão preparados para alterar os parâmetros que condicionam o problema, através de um mecanismo simples de troca de ficheiros, os programas PROAES_FORM e PROAES podem ser empregues para realizar a optimização de estruturas.

Para isso é necessário dispor de um algoritmo de optimização. Os vários algoritmos disponíveis na ferramenta de optimização do MATLAB, *optimtool*, podem ser utilizados com esta finalidade. Nesta dissertação utilizou-se o algoritmo programação sequencial quadrática (SQP).

Quando o problema de optimização é determinístico, todas as funções usadas na formulação do problema podem ser calculadas através do programa PROAES, este caso está representado na Figura 5.3.

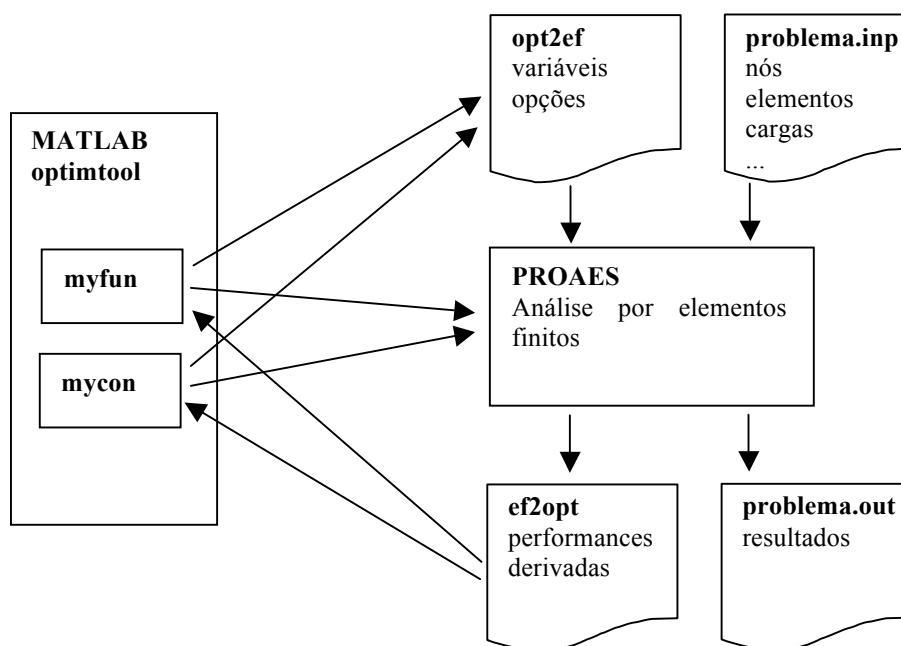


Figura 5.3 - Diagrama de execução para um problema de otimização determinístico.

Quando algumas variáveis são aleatórias, os programas PROAES e PROAES_FORM, funcionam em conjunto com a *optimtool*, por forma a efectuar optimização, Figura 5.4. Neste caso, todos os ficheiros associados aos programas PROAES e PROAES_FORM devem ser definidos, para que a troca de ficheiros entre módulos seja fluída e completa.

Com os programas desenvolvidos é possível efectuar-se RBDO e RDO. A formulação do problema é definida nos ficheiros *myfunc.m* e *mycon.m*, onde deverão estar as chamadas ao PROAES e ao PROAES_FORM necessárias ao cálculo da função objectivo e ao cálculo dos constrangimentos, respectivamente. O mecanismo de troca de informação através de ficheiros desenvolvido exige que seja escrito o ficheiro **opt2form** ou **opt2ef** antes de executar o PROAES_FORM ou o PROAES e de ler os correspondentes **form2opt** ou **ef2opt**.

Os ficheiros que definem a função objectivo e os constrangimentos, são escritos em linguagem MATLAB e lidos pela *optimtool*, que procede à optimização utilizando um algoritmo acoplado, Figura 5.4.

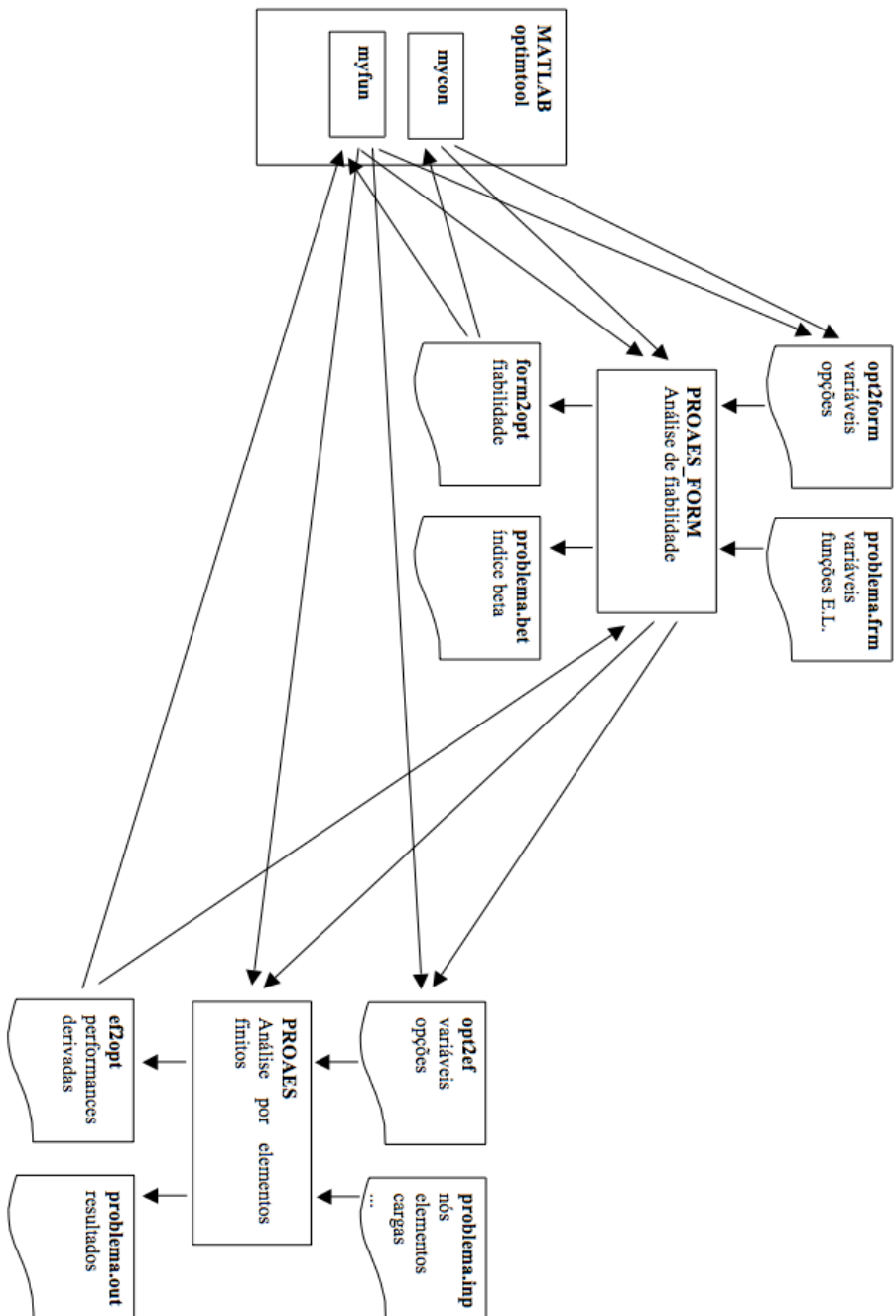


Figura 5.4 - Diagrama de execução para um problema de otimização probabilístico.

Capítulo 6 - Exemplos

6.1 Introdução

De modo a validar os resultados obtidos pelos programas desenvolvidos, efectuaram-se diversas análises e comparações com resultados apresentados por outros autores. Estudaram-se 4 exemplos. Os exemplos 1 e 3 foram retirados de Cardoso [36], o exemplo 2 é apresentado no programa FERUM [13] e o exemplo 4 foi apresentado por Stocki et al. [56].

Inicialmente é importante saber quais as características do problema que devem ser tomadas como relevantes para a sua resolução. Essas características são escritas de uma forma conveniente nos ficheiros de dados, que serão lidos pelos programas.

Sobre o modelo de elementos finitos da estrutura importa saber o número de nós, de elementos, bem como as dimensões que a caracterizam, assim como os apoios que a constituem. É importante também saber, os módulos de elasticidade (E), os momentos de inércia (I) e as áreas da secção transversal (A) de cada elemento. Dados que também requerem uma especial atenção são as forças e/ou cargas distribuídas aplicadas nos nós e os respectivos valores.

É necessário identificar o número de variáveis aleatórias presentes no problema e para cada uma delas é indispensável saber a sua distribuição probabilística, média e desvio padrão.

É também necessário definir as funções de estado limite que serão usadas na avaliação da segurança da estrutura.

6.2 Exemplo 1 – Pórtico

O primeiro exemplo apresentado foi retirado de Cardoso [36], e está ilustrado na Figura 6.1. Pretendeu-se validar as sensibilidades calculadas pelo PROAES, comparando-as com a referência [36] e com os resultados obtidos com diferenças finitas progressivas.

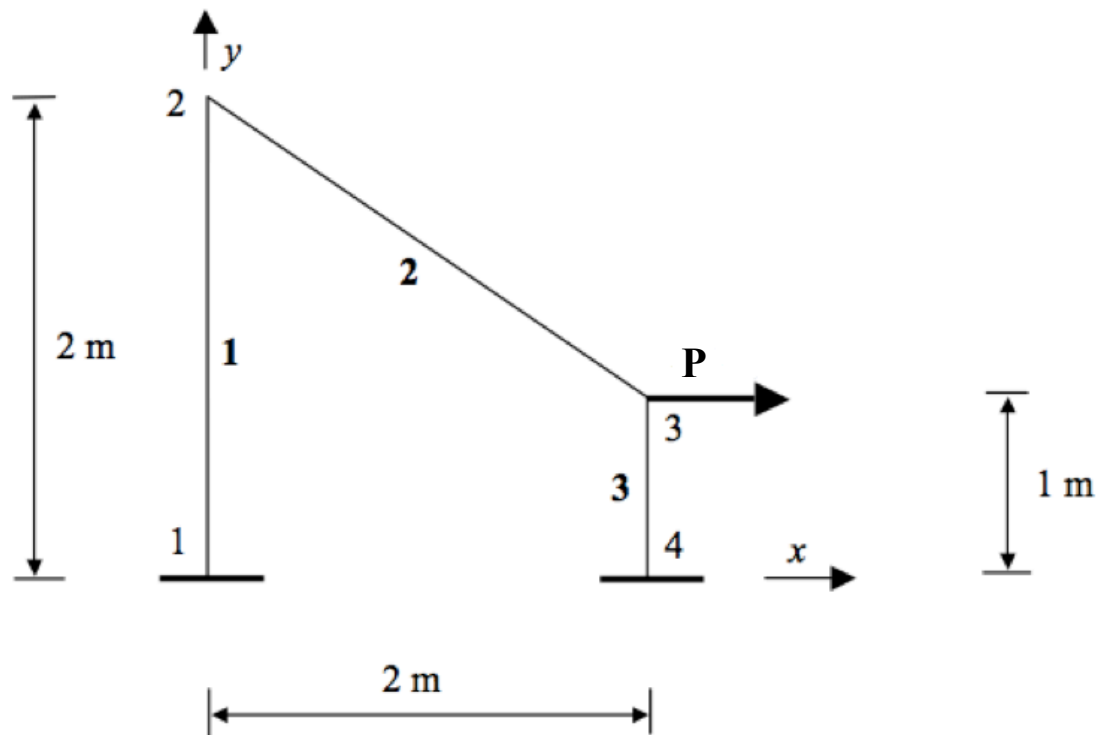


Figura 6.1 - Pórtico, exemplo 1.

Dados do problema:

$$E = 210 \text{ GPa}$$

$$D = 20 \text{ mm}$$

$$P = 200 \text{ N}$$

O problema consiste num pórtico composto por 4 nós e 3 elementos. Estes têm secção transversal circular com diâmetro D . As vigas que constituem o pórtico são construídas com um material de módulo de Young, E .

Neste exemplo considerou-se que existem 5 performances e 6 variáveis de projecto descritas nas Tabelas 6.1 e 6.2, respectivamente.

Tabela 6.1 - Performances do exemplo 1.

Performance	Descrição	Valor
1	Deslocamento do nó 2 segundo X	$+1.686239 \times 10^{-2}$ m
2	Deslocamento do nó 3 segundo X	$+1.686255 \times 10^{-2}$ m
3	Deslocamento do nó 2 segundo Y	$+7.894123 \times 10^{-7}$ m
4	Deslocamento do nó 3 segundo Y	-3.947061×10^{-7} m
5	Volume	$+1.644958 \times 10^{-3}$ m ³

Tabela 6.2 - Variáveis de projecto do exemplo 1.

Variável	Descrição	Valor
1	Coordenada X do nó 2	0.0 m
2	Coordenada Y do nó 2	2.0 m
3	Módulo de elasticidade do material	210 GPa
4	Factor de carga	1
5	Área da secção transversal	3.14159×10^{-4} m ²
6	Inércia da secção transversal	7.85398×10^{-9} m ⁴

Efectuou-se o cálculo das sensibilidades pelo método contínuo, Tabela 6.3, e pelo método da diferenças finitas com $\Delta = 0.1\%$, Tabela 6.4. Como se pode constatar os valores são bastante consistentes quando feita a comparação das duas tabelas.

Tabela 6.3 - Sensibilidades do exemplo 1 (calculadas pelo método contínuo).

Var./Perf.	1	2	3	4	5
1	-8.437189×10^{-3}	-4.222351×10^{-3}	-8.430792×10^{-3}	-2.023638×10^{-7}	-2.809924×10^{-4}
2	5.244348×10^{-3}	5.242958×10^{-3}	2.605855×10^{-7}	6.706032×10^{-8}	4.546552×10^{-4}
3	$-8.029711 \times 10^{-14}$	$-8.029787 \times 10^{-14}$	$-3.759106 \times 10^{-18}$	1.879553×10^{-18}	0
4	1.686239×10^{-2}	1.686255×10^{-2}	7.894123×10^{-7}	-3.947061×10^{-7}	0
5	-1.974136×10^{-4}	-7.024672×10^{-4}	-2.512749×10^{-3}	1.256375×10^{-3}	5.236068
6	$-2.146979 \times 10^{+6}$	$-2.146979 \times 10^{+6}$	-1.213332×10^{-3}	6.066661×10^{-4}	0

Tabela 6.4 - Sensibilidades do exemplo 1 (calculadas por diferenças finitas , $\Delta = 0,1\%$ do valor de cada variável).

Var./Perf.	1	2	3	4	5
1	-8.436969×10^{-3}	-4.222695×10^{-3}	-8.426040×10^{-3}	-2.024132×10^{-7}	-2.808997×10^{-4}
2	5.241420×10^{-3}	5.239439×10^{-3}	2.605963×10^{-7}	6.698792×10^{-8}	4.547676×10^{-4}
3	$-8.021690 \times 10^{-14}$	$-8.021765 \times 10^{-14}$	$-3.755351 \times 10^{-18}$	1.877675×10^{-18}	0
4	1.686239×10^{-2}	1.686255×10^{-2}	7.894123×10^{-7}	-3.947061×10^{-7}	0
5	-1.973523×10^{-4}	-7.002823×10^{-4}	-2.510239×10^{-3}	1.255119×10^{-3}	5.236068
6	$-2.144834 \times 10^{+6}$	$-2.144834 \times 10^{+6}$	-1.213398×10^{-3}	6.073359×10^{-4}	0

6.3 Exemplo 2 – Pórtico de 2 andares

Como segundo exemplo, é utilizado um dos problemas apresentado no programa FERUM [13].

Neste exemplo será efectuada uma análise de fiabilidade, em que são usados os programas PROAES e PROAES_FORM. O programa PROAES calcula a resposta da estrutura e as suas derivadas, que servem para definir as funções de estado limite. O programa PROAES_FORM calcula os índices β associados a uma ou várias funções de estado limite que condicionam a segurança da estrutura.

Na Figura 6.2 está representado o problema, um pórtico de 2 andares composto por 9 nós e 10 elementos.

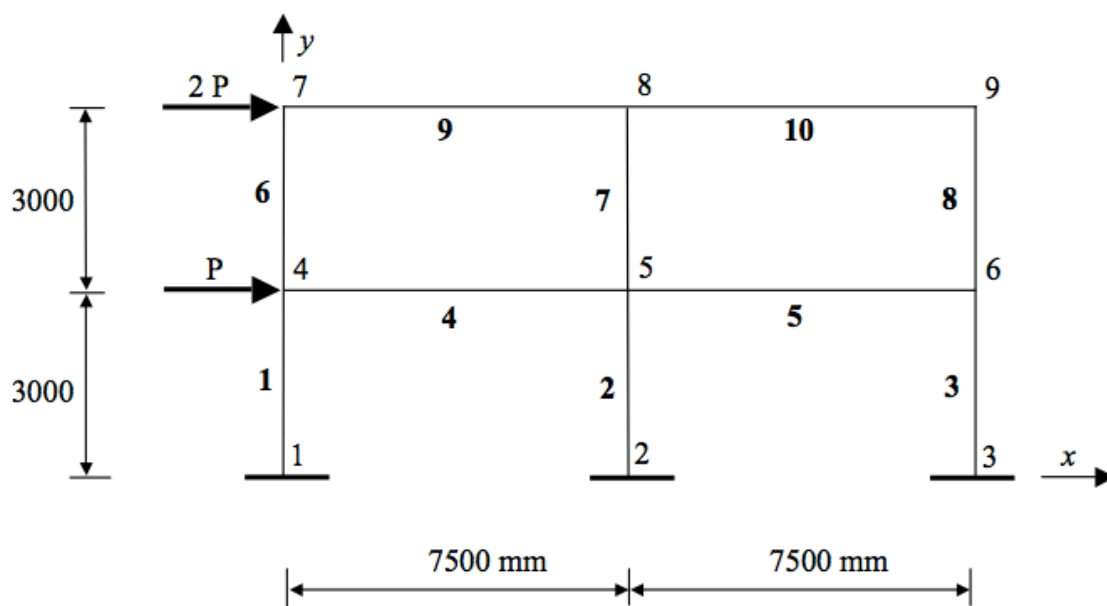


Figura 6.2 - Pórtico de 2 andares, exemplo 2.

Dados do problema:

$$E = 200000 \text{ N/mm}^2$$

$$A = 5000 \text{ mm}^2$$

$$I = 0.500 \times 10^9 \text{ mm}^4$$

$$P = 300 \text{ kN}$$

Neste exemplo foram consideradas 2 performances e 31 variáveis de projecto.

Tabela 6.5 - Performances do exemplo 2.

Performance	Descrição	Valor
1	Volume	$+2.40 \times 10^8 \text{ m}^2$
2	Deslocamento do nó 9 segundo X	+30.42760 mm

Para determinar a fiabilidade do pórtico, considera-se que todas as 31 variáveis definidas no ficheiro **portico_2andares.inp** [Anexo 3] são aleatórias. Assume-se que o módulo de elasticidade, E , o momento de inércia, I , e a área da secção transversal, A , de cada um dos 10 elementos são variáveis independentes, estando agrupadas da seguinte forma: as 10 primeiras correspondem ao módulo de elasticidade; as 10 seguintes ao momento de inércia e as 10 seguintes às áreas. Todas estas 30 variáveis são caracterizadas por ter uma distribuição normal, e as médias e desvios padrão indicados. A variável 31 corresponde ao multiplicador das cargas aplicadas, P , que tem uma distribuição lognormal, com a média e o desvio padrão indicado. A Tabela 6.6 mostra a média e o desvio padrão de cada um dos tipos das 31 variáveis de projecto apresentadas na Tabela 6.7.

Tabela 6.6 - Média e desvio padrão de cada tipo de variável, exemplo 2.

Variável	Média	Desvio
E	200000 N/mm ²	20000 N/mm ²
I	$0.500 \times 10^9 \text{ mm}^4$	$0.500 \times 10^8 \text{ mm}^4$
A	5000 mm ²	500 mm ²
P	300 kN	45 kN

Considerou-se uma única função de estado limite, $g(X) = \delta_{Adm} - \delta_{Nó}$, baseado no deslocamento horizontal do nó 9. Foi considerado o valor admissível do deslocamento igual a 45 mm.

Tabela 6.7 - Variáveis de projecto do exemplo 2.

Variável	Descrição	Valor
1	Módulo de elasticidade do elemento 1	200000 N/mm ²
2	Módulo de elasticidade do elemento 2	200000 N/mm ²
3	Módulo de elasticidade do elemento 3	200000 N/mm ²
4	Módulo de elasticidade do elemento 4	200000 N/mm ²
5	Módulo de elasticidade do elemento 5	200000 N/mm ²
6	Módulo de elasticidade do elemento 6	200000 N/mm ²
7	Módulo de elasticidade do elemento 7	200000 N/mm ²
8	Módulo de elasticidade do elemento 8	200000 N/mm ²
9	Módulo de elasticidade do elemento 9	200000 N/mm ²
10	Módulo de elasticidade do elemento 10	200000 N/mm ²
11	Inércia da secção do elemento 1	0.500×10 ⁹ mm ⁴
12	Inércia da secção do elemento 2	0.500×10 ⁹ mm ⁴
13	Inércia da secção do elemento 3	0.500×10 ⁹ mm ⁴
14	Inércia da secção do elemento 4	0.500×10 ⁹ mm ⁴
15	Inércia da secção do elemento 5	0.500×10 ⁹ mm ⁴
16	Inércia da secção do elemento 6	0.500×10 ⁹ mm ⁴
17	Inércia da secção do elemento 7	0.500×10 ⁹ mm ⁴
18	Inércia da secção do elemento 8	0.500×10 ⁹ mm ⁴
19	Inércia da secção do elemento 9	0.500×10 ⁹ mm ⁴
20	Inércia da secção do elemento 10	0.500×10 ⁹ mm ⁴
21	Área da secção do elemento 1	5000 mm ²
22	Área da secção do elemento 2	5000 mm ²
23	Área da secção do elemento 3	5000 mm ²
24	Área da secção do elemento 4	5000 mm ²
25	Área da secção do elemento 5	5000 mm ²
26	Área da secção do elemento 6	5000 mm ²
27	Área da secção do elemento 7	5000 mm ²
28	Área da secção do elemento 8	5000 mm ²
29	Área da secção do elemento 9	5000 mm ²
30	Área da secção do elemento 10	5000 mm ²
31	Factor de carga P	300000

O PROAES_FORM produziu os seguintes resultados:

Diferenças Finitas	Método Contínuo
<pre>>> PROAES_FORM('portico_ferum') PROAES_FORM - Iteration 1 ----- ***** beta(1)= 3.108509 - Iteration 2 ----- ***** beta(2)= 2.574266 - Iteration 3 ----- ***** beta(3)= 2.550319 - Iteration 4 ----- ***** beta(4)= 2.550273 ----- Final beta= 2.550273 Corresponding Pf= 0.00538195 No Iterations= 4 , No Functions Eval= 133</pre>	<pre>>> PROAES_FORM('portico_ferum') PROAES_FORM - Iteration 1 ----- ***** beta(1)= 3.108508 - Iteration 2 ----- ***** beta(2)= 2.574266 - Iteration 3 ----- ***** beta(3)= 2.550319 - Iteration 4 ----- ***** beta(4)= 2.550273 ----- Final beta= 2.550273 Corresponding Pf= 0.00538195 No Iterations= 4 , No Functions Eval= 9</pre>

Figura 6.3 - Resultados PROAES_FORM, exemplo 2.

A Tabela 6.8 mostra a comparação dos resultados provenientes do programa FERUM e do PROAES_FORM. Como pode ser observado os resultados obtidos pelo PROAES_FORM foram bastante precisos quando comparados com os resultados do FERUM.

Tabela 6.8 - Comparação de resultados do exemplo 2.

	FERUM	PROAES_FORM
P_f	5.381×10^{-3}	5.382×10^{-3}
β	2.550	2.550

O MATLAB tem uma função chamada PROFILE onde é possível visualizar qual o tempo dispendido em cada função. Esta utilidade foi accionada no PROAES_FORM desde o início da resolução do algoritmo FORM até ser atingida a convergência. Assim, o exemplo mencionado foi resolvido quer pelo método de diferenças finitas quer pelo método contínuo, de modo a calcular o tempo dispendido em ambos os casos, permitindo assim fazer uma análise comparativa.

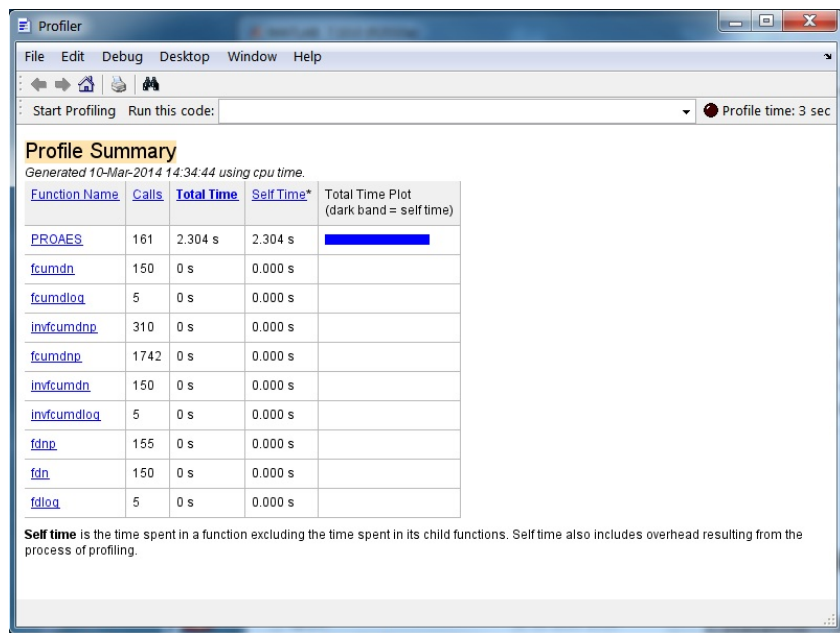


Figura 6.4 - Sumário PROFILE, exemplo 2 (diferenças finitas).

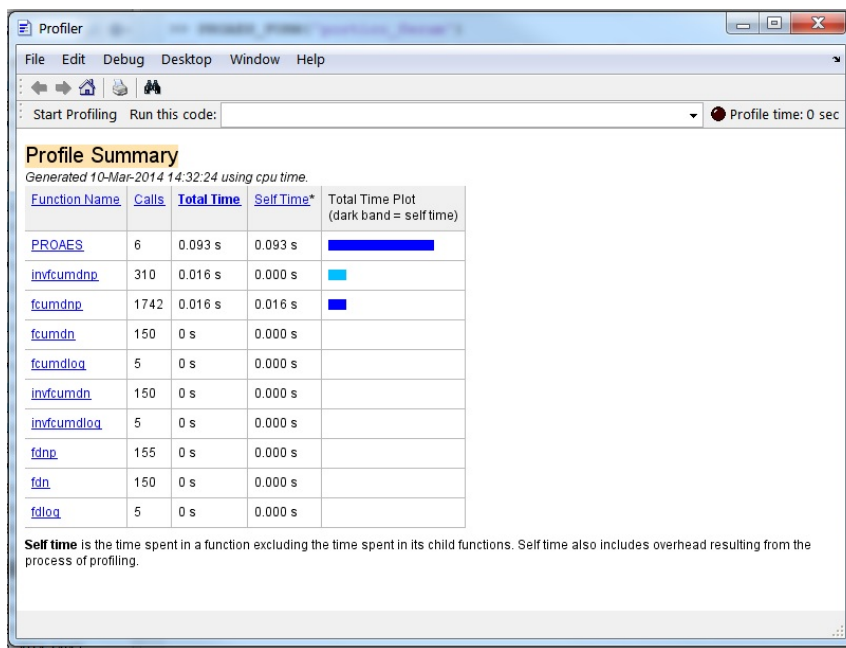


Figura 6.5 - Sumário PROFILE, exemplo 2 (método contínuo).

Considerando apenas o tempo despendido em análises pelo PROAES, que é muito superior ao usado pelo algoritmo FORM, as 161 análises estruturais por elementos finitos, necessárias para cálculo da função de estado limite e suas derivadas utilizando diferenças finitas requereram 2.304 s.

Comparativamente, as 6 análises pelo método dos elementos finitos juntamente com as 6 análises de sensibilidade pelo método contínuo demoraram 0.093 s o que corresponde a um ganho de 96%.

O PROAES_FORM também calcula a derivada do índice de fiabilidade, β , em ordem aos dois parâmetros usados para caracterizar a distribuição estatística de cada variável, o valor médio, $\mu_{(x)}$, e o desvio padrão, $\sigma_{(x)}$. Para este problema, obtiveram-se os resultados apresentados nas Tabelas 6.9 e 6.10 que são comparados com os obtidos por diferenças finitas considerando uma perturbação $\Delta = 0.1\%$ do valor de cada parâmetro.

Tabela 6.9 - Derivadas de β em ordem aos valores médios das variáveis, calculadas no ponto X* (exemplo 2).

Variável	$d\beta / d\mu_{(X)}(\text{PROAES FORM})$	$d\beta / d\mu_{(X)}(\text{diferenças finitas})$
1	3.858348×10^{-6}	3.857202×10^{-6}
2	4.330549×10^{-6}	4.328995×10^{-6}
3	3.371611×10^{-6}	3.370686×10^{-6}
4	5.574401×10^{-6}	5.572593×10^{-6}
5	5.638641×10^{-6}	5.636811×10^{-6}
6	9.694834×10^{-6}	9.686610×10^{-6}
7	3.266974×10^{-6}	3.264301×10^{-6}
8	1.116377×10^{-6}	1.115415×10^{-6}
9	1.973086×10^{-6}	1.972220×10^{-6}
10	2.234779×10^{-6}	2.233774×10^{-6}
11	1.488325×10^{-9}	1.487920×10^{-9}
12	1.732695×10^{-9}	1.732073×10^{-9}
13	1.292111×10^{-9}	1.291796×10^{-9}
14	2.272857×10^{-9}	2.272111×10^{-9}
15	2.265952×10^{-9}	2.265209×10^{-9}
16	3.787172×10^{-10}	3.783954×10^{-10}
17	1.306909×10^{-9}	1.305840×10^{-9}
18	4.366532×10^{-10}	4.362763×10^{-10}
19	1.104771×10^{-9}	1.104177×10^{-9}
20	1.212264×10^{-9}	1.211617×10^{-9}
21	5.292301×10^{-6}	5.287049×10^{-6}
22	-4.541452×10^{-8}	-4.537046×10^{-8}
23	5.466826×10^{-6}	5.461399×10^{-6}
24	-4.060070×10^{-6}	-4.056906×10^{-6}
25	-9.889584×10^{-7}	-9.881914×10^{-7}
26	8.988443×10^{-7}	8.979493×10^{-7}
27	-1.154720×10^{-8}	-1.153579×10^{-8}
28	9.787412×10^{-7}	9.777670×10^{-7}
29	-3.067164×10^{-5}	-3.064267×10^{-5}
30	-3.085883×10^{-5}	-3.082970×10^{-5}
31	-1.407457×10^{-5}	-1.408271×10^{-5}

Tabela 6.10 - Derivadas de β em ordem aos desvios padrões das variáveis, calculadas no ponto X* (exemplo 2).

Variável	$d\beta / d\sigma_{(X)} \text{ (PROAES FORM)}$	$d\beta / d\beta / d\sigma_{(X)} \text{ (diferenças finitas)}$
1	-7.593109×10^{-7}	-7.597015×10^{-7}
2	-9.565392×10^{-7}	-9.570370×10^{-7}
3	-5.798181×10^{-7}	-5.801149×10^{-7}
4	-1.584942×10^{-6}	-1.585763×10^{-6}
5	-1.621682×10^{-6}	-1.622522×10^{-6}
6	-4.793992×10^{-8}	-4.796467×10^{-8}
7	-5.443873×10^{-7}	-5.446860×10^{-7}
8	-6.356797×10^{-8}	-6.360103×10^{-8}
9	-1.985677×10^{-7}	-1.986700×10^{-7}
10	-2.547334×10^{-7}	-2.548651×10^{-7}
11	$-2.824571 \times 10^{-10}$	$-2.826018 \times 10^{-10}$
12	$-3.828256 \times 10^{-10}$	$-3.830248 \times 10^{-10}$
13	$-2.128905 \times 10^{-10}$	$-2.129990 \times 10^{-10}$
14	$-6.587201 \times 10^{-10}$	$-6.590616 \times 10^{-10}$
15	$-6.547242 \times 10^{-10}$	$-6.550636 \times 10^{-10}$
16	$-1.828886 \times 10^{-11}$	$-1.829833 \times 10^{-11}$
17	$-2.177947 \times 10^{-10}$	$-2.179142 \times 10^{-10}$
18	$-2.431251 \times 10^{-11}$	$-2.432514 \times 10^{-11}$
19	$-1.556328 \times 10^{-10}$	$-1.557146 \times 10^{-10}$
20	$-1.873920 \times 10^{-10}$	$-1.874908 \times 10^{-10}$
21	-3.571459×10^{-8}	-3.573196×10^{-8}
22	$-2.629941 \times 10^{-12}$	$-2.416201 \times 10^{-12}$
23	-3.810896×10^{-8}	-3.813012×10^{-8}
24	-2.101957×10^{-8}	-2.102845×10^{-8}
25	-1.247132×10^{-9}	-1.248275×10^{-9}
26	-1.030210×10^{-9}	-1.034008×10^{-9}
27	$-1.700238 \times 10^{-13}$	$-1.914824 \times 10^{-13}$
28	-1.221497×10^{-9}	-1.224121×10^{-9}
29	-1.199584×10^{-6}	-1.200166×10^{-6}
30	-1.214271×10^{-6}	-1.214860×10^{-6}
31	-4.694990×10^{-5}	-4.690351×10^{-5}

6.4 Exemplo 3 – Treliça de 31 barras

No exemplo 3 efectuar-se-á RBDO e RDO numa estrutura composta por 31 elementos. A estrutura é mostrada na Figura 6.6 na configuração inicial, e é constituída por seis partes iguais, com 3 m de comprimento e 5 metros de altura, com uma extensão total de 18 m.

Os elementos são agrupados em três conjuntos, e a cada um desses conjuntos corresponde a mesma secção transversal. A Tabela 6.11 mostra a composição dos grupos.

O modelo estocástico do problema é apresentado detalhadamente na Tabela 6.12. Neste caso, foram utilizadas 12 variáveis básicas, que incluem as áreas das secções transversais de cada grupo, coordenadas Y de sete nós distintos, módulo de elasticidade e o factor de carga.

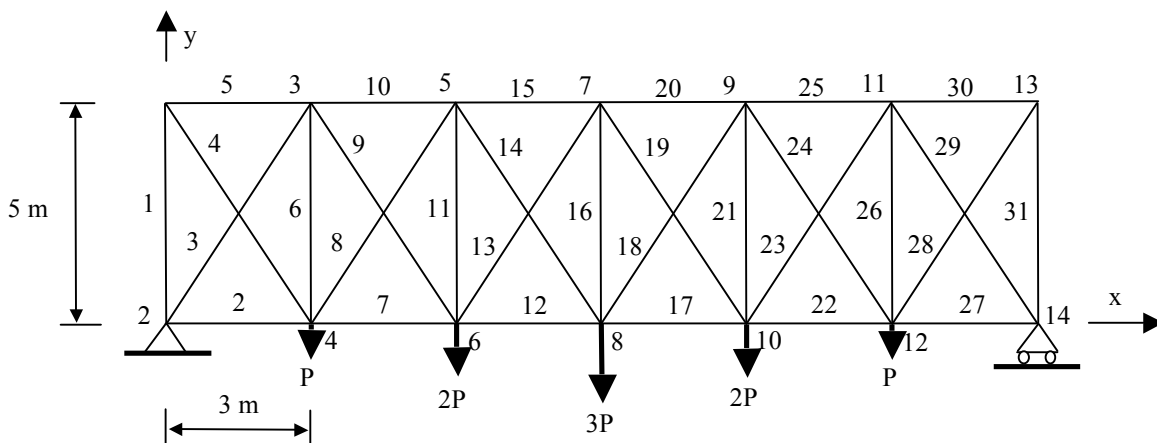


Figura 6.6 - Configuração inicial do exemplo 3 [36].

Para todas as cargas aplicadas mostradas na Figura 6.6, $P = 1\text{ MN}$.

Tabela 6.11 - Grupos de elementos do exemplo 3.

Grupo	Elementos
1	2, 7, 12, 17, 22, 27
2	3, 10, 15, 20, 25, 29
3	1, 4, 5, 6, 8, 9, 11, 13, 14, 16, 18, 19, 21, 23, 24, 26, 28, 30, 31

Tabela 6.12 - Modelo estocástico do exemplo 3.

Var.	Descrição	Distribuição	Média	Desvio Padrão
1	Área da secção transversal, 1	Normal	0.0314159 m ²	0.00314159 m ²
2	Área da secção transversal, 2	Normal	0.0314159 m ²	0.00314159 m ²
3	Área da secção transversal, 3	Normal	0.0314159 m ²	0.00314159 m ²
4	Nó 1 coordenada Y	Normal	5 m	0.05 m
5	Nó 3 coordenada Y	Normal	5 m	0.05 m
6	Nó 5 coordenada Y	Normal	5 m	0.05 m
7	Nó 7 coordenada Y	Normal	5 m	0.05 m
8	Nó 9 coordenada Y	Normal	5 m	0.05 m
9	Nó 11 coordenada Y	Normal	5 m	0.05 m
10	Nó 13 coordenada Y	Normal	5 m	0.05 m
11	Módulo de Young	Normal	210 GPa	0.105 GPa
12	Factor de carga	Gumbel	1.0	0.25

Em seguida definiu-se o conjunto de performances que caracterizam o comportamento estrutural e que permitem definir a função objectivo, os constrangimentos e as funções de estado limite (LSF). Neste exemplo, o volume total é considerado um desempenho estrutural fundamental que deve ser optimizado. O deslocamento vertical do nó 8 também deve ser calculado, pois este valor deve ser limitado. Como os 31 elementos são agrupados em apenas 3 conjuntos, a tensão nos elementos de maior tensão de cada conjunto deve ser obtida. A Tabela 6.13 mostra todas as performances definidas, com os valores iniciais correspondentes.

O programa desenvolvido deve calcular as derivadas de cada uma das performances na Tabela 6.13 no que diz respeito às variáveis na Tabela 6.12.

Tabela 6.13 – Performances do exemplo 3.

Performance	Descrição	Valor inicial
1	Volume	+4.428744 m ³
2	Deslocamento vertical do nó 8	-1.730663×10 ⁻² m
3	Tensão no elemento 9	+7.6247×10 ⁷ Pa
4	Tensão no elemento 12	+1.6498×10 ⁸ Pa
5	Tensão no elemento 15	-1.7782×10 ⁸ Pa
6	Tensão no elemento 17	+1.6498×10 ⁸ Pa
7	Tensão no elemento 20	-1.7782×10 ⁸ Pa
8	Tensão no elemento 23	+7.6247×10 ⁷ Pa

Associadas às últimas sete performances foram estabelecidas sete funções de estado limite (LSF). A Tabela 6.14 mostra os índices de fiabilidade (calculados pelo FORM) de cada LSF, para a configuração inicial na Figura 6.6. A primeira LSF é definida como $g(X) = \delta_{Adm} + \delta_{Nó}$, a 4ª e 6ª por $g(X) = \sigma_{Adm} + \sigma_{Elemento}$, pois as performances correspondentes têm valores negativos, as restantes LSF são definidas por $g(X) = \sigma_{Adm} - \sigma_{Elemento}$.

Considerou-se o deslocamento admissível $\delta_{Adm} = L / 500 = 36mm$ e uma tensão admissível $\sigma_{Adm} = 350MPa$.

Tabela 6.14 - LSF e índices de fiabilidade do exemplo 3, para configuração inicial.

LSF	Performance	Descrição	β
1	2	Deslocamento vertical do nó 8	2.7899
2	3	Tensão no elemento 9	5.1720
3	4	Tensão no elemento 12	2.7594
4	5	Tensão no elemento 15	2.5187
5	6	Tensão no elemento 17	2.7594
6	7	Tensão no elemento 20	2.5187
7	8	Tensão no elemento 23	5.1720

6.4.1 RBDO

Tendo estabelecido o modelo estocástico, performances e LSF, um problema RBDO clássico é então formulado.

O volume total é a função objectivo a ser minimizada com restrições sobre os índices de fiabilidade, que devem ser todos maiores do que $\beta_{Min} = 3.7$. Um total de 10 variáveis de projeto são utilizadas, três estão associadas aos valores médios das variáveis básicas 1 a 3 da Tabela 6.12, que corresponde às áreas das secções transversais e as outras são associadas aos valores médios das coordenadas dos nós 1, 3, 5, 7, 9, 11 e 13. Todas as variáveis de projeto são apresentadas na Tabela 6.15, com os valores iniciais e limites inferior e superior.

A fim de resolver o problema com os programas desenvolvidos, os ficheiros de entrada que descrevem a análise de elementos finitos e o modelo estocástico são criados. Também devem ser desenvolvidas duas funções MATLAB, para calcular a função

objectivo e os constrangimentos. A *optimtool* do MATLAB é usada com o algoritmo de programação quadrática sequencial (SQP).

Tabela 6.15 - Variáveis de projecto para as formulações RBDO e RDO do exemplo 3.

Variável Projecto	Descrição	Valor inicial	Limite inferior	Limite superior
1	Área da secção transversal, 1	0.0314159 m ²	0.0113097 m ²	0.0804248 m ²
2	Área da secção transversal, 2	0.0314159 m ²	0.0113097 m ²	0.0804248 m ²
3	Área da secção transversal, 3	0.0314159 m ²	0.0113097 m ²	0.0804248 m ²
4	Nó 1 coordenada Y	5 m	3.5 m	6.0 m
5	Nó 3 coordenada Y	5 m	3.5 m	6.0 m
6	Nó 5 coordenada Y	5 m	3.5 m	6.0 m
7	Nó 7 coordenada Y	5 m	3.5 m	6.0 m
8	Nó 9 coordenada Y	5 m	3.5 m	6.0 m
9	Nó 11 coordenada Y	5 m	3.5 m	6.0 m
10	Nó 13 coordenada Y	5 m	3.5 m	6.0 m

A convergência é obtida em 18 iterações, como mostrado na Figura 6.7. A Tabela 6.16 apresenta os valores iniciais e finais obtidos para as variáveis de projecto, a função objectivo e os constrangimentos.

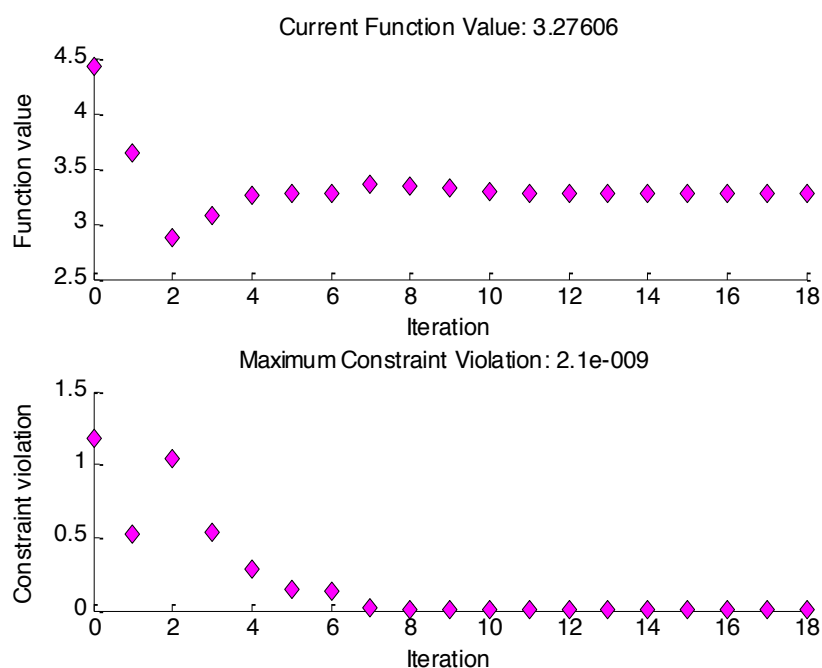


Figura 6.7 - Convergência para o problema RBDO, exemplo 3.

Para este problema, o tempo total de optimização é de apenas 166 s e requer 42 chamadas para que o programa calcule a função objectivo, constrangimentos e gradientes.

Alterou-se o programa PROAES_FORM de forma a utilizar diferenças finitas, neste caso o tempo total de optimização é de 822 s, a diferença de tempo deve-se ao facto do método das diferenças finitas necessitar de calcular mais vezes a função de estado limite.

A Figura 6.8 representa a configuração da estrutura optimizada.

Tabela 6.16 - Valores iniciais e finais para o problema RBDO, exemplo 3.

Descrição	Inicial	Final
Var. Projecto 1 (m ²)	0.0314159	0.0393442
Var. Projecto 2 (m ²)	0.0314159	0.0514117
Var. Projecto 3 (m ²)	0.0314159	0.0148589
Var. Projecto 4 (m)	5.0	3.50000
Var. Projecto 5 (m)	5.0	3.63230
Var. Projecto 6 (m)	5.0	5.54038
Var. Projecto 7 (m)	5.0	6.00000
Var. Projecto 8 (m)	5.0	5.54038
Var. Projecto 9 (m)	5.0	3.63230
Var. Projecto 10 (m)	5.0	3.50000
Volume (m ³)	4.42874	3.27606
β_1	2.7899	3.7000
β_2	5.1720	3.7000
β_3	2.7594	4.1761
β_4	2.5187	4.6250
β_5	2.7594	4.1761
β_6	2.5187	4.6250
β_7	5.1720	3.7000

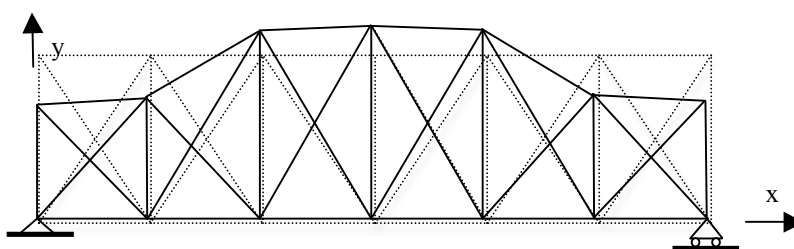


Figura 6.8 - Configuração final da estrutura sobreposta à inicial, obtida pela formulação RBDO (exemplo 3).

6.4.2 RDO

Neste caso, considerou-se a primeira LSF crítica e a variância de β_1 calculada pela equação (6.1), com os gradientes β fornecidos pelo programa. Esta LSF foi escolhido por estar sempre activa durante o processo de optimização e é considerada como uma restrição crítica durante o tempo de vida da estrutura. Considerando todas as variáveis fixas nos valores óptimos, exceto uma, um gráfico da variação de β_1 com esta variável é mostrado na Figura 6.9 para as primeiras quatro variáveis de projeto.

$$Var\beta = \sum_{i=1}^{NV} \left(\frac{\partial \beta}{\partial x_i} \right)^2 \cdot \sigma_{x_i}^2 \quad (6.1)$$

Para obter a Figura 6.9, o valor médio da variável seleccionada toma 21 valores igualmente espaçados entre os limites superior e inferior definidos no problema RBDO. O desvio padrão é mantido constante e igual ao inicial, Tabela 6.12.

Observando a Figura 6.9, é evidente que a variância é não linear. Uma estrutura optimizada deve ter baixo peso e alta fiabilidade. Mas é requerido que mantenha um nível de segurança constante na presença de acções aleatórias.

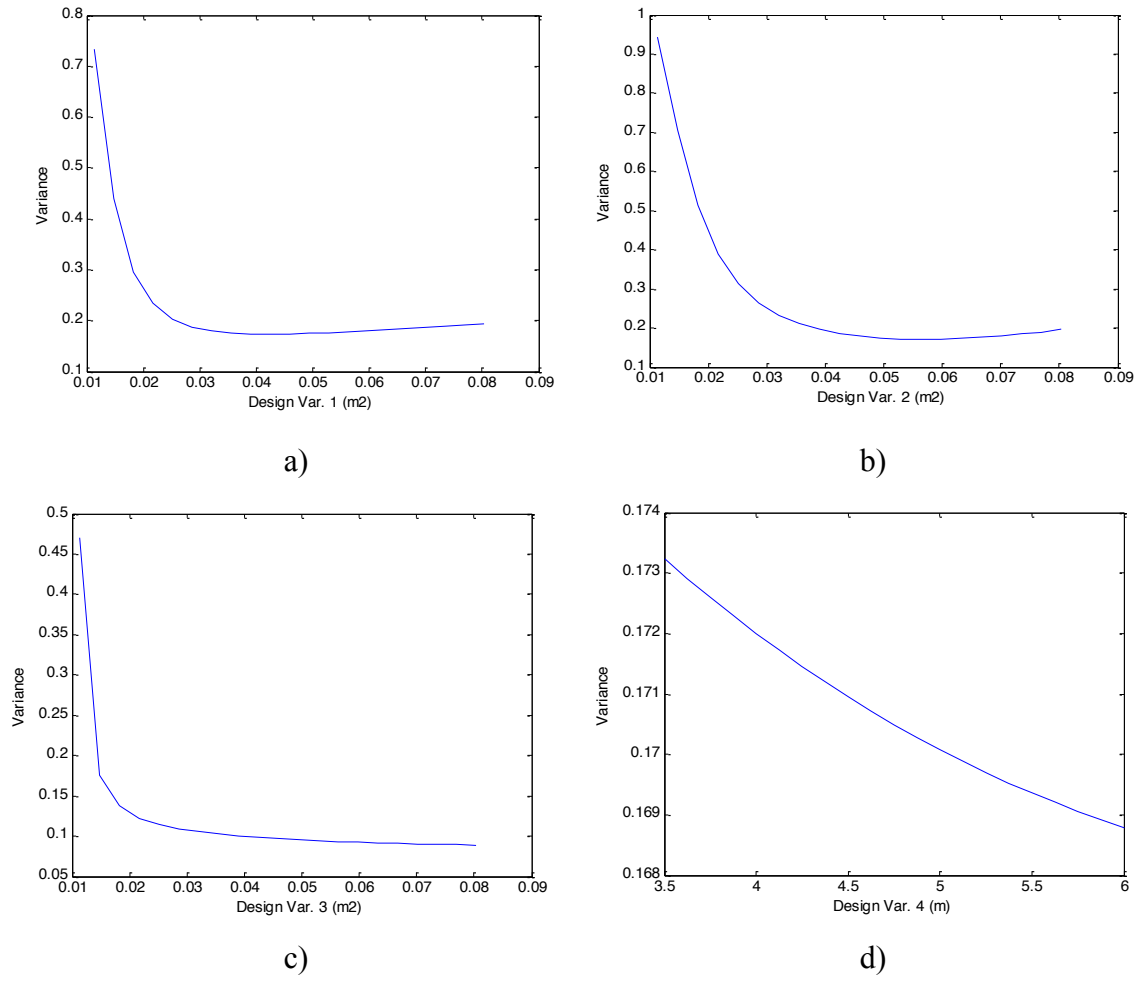


Figura 6.9 - Gráfico da variância de β_1 com as variáveis de projecto de 1 a 4, respectivamente de a) a d), (exemplo 3).

Como já mencionado anteriormente, o conceito de robustez está associado à menor variabilidade da resposta estrutural no que diz respeito às incertezas do projeto. Formulou-se o problema RDO, onde a função objectivo é tida como uma soma ponderada do volume estrutural e da variância, equação (6.2), isto é, o problema de optimização é agora multiobjectivo.

$$f = (1 - \alpha) \frac{vol}{c_1} + \alpha \frac{var[\beta]}{c_2} \quad (6.2)$$

onde vol e $var[\beta]$ representam o volume total e a variância β_1 , respectivamente, c_1 e c_2 são duas constantes que normalizam os dois termos da equação e α é um coeficiente que permite definir a importância relativa desses termos. Para o presente estudo $c_1 = 3.0$ e $c_2 = 0.15$

Variando-se o valor de α no intervalo $[0;0.8]$, é possível obter a frente de Pareto, Figura 6.10, de soluções não-dominadas ótimas mostradas na Tabela 6.17.

Tabela 6.17 - Volume e Variância em função de α , exemplo 3.

α	Volume (m ³)	Variância
0	3.276055	0.173239
0.1	3.278578	0.171213
0.2	3.315510	0.160389
0.3	3.462084	0.150936
0.4	3.516162	0.146287
0.5	3.918019	0.121982
0.6	4.524115	0.098191
0.7	5.281815	0.078278
0.8	6.040410	0.066526

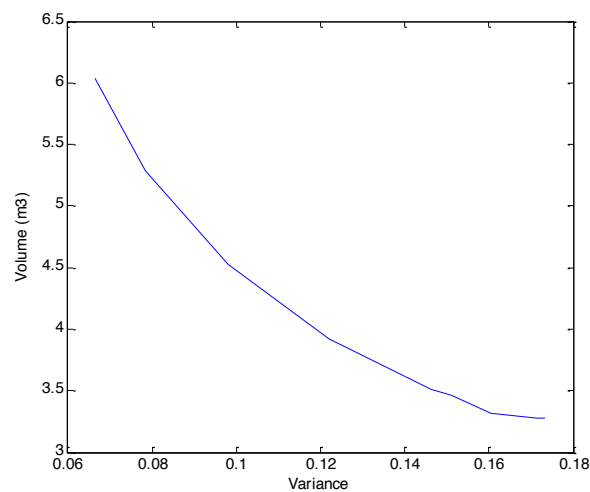


Figura 6.10 - Frente de Parteo das soluções ótimas para a RDO, exemplo 3.

Claramente menores valores de variância não têm interesse para o estudo, pois estão associados a volumes estruturais muito elevados. Como os constrangimentos impostos foram apenas nos limites inferiores de β , as soluções otimizadas para valores elevados de α são muito seguras, com um alto índice de fiabilidade, como mostra a Tabela 6.18, em que a solução otimizada para $\alpha = 0.8$ é apresentada.

Tabela 6.18 - Valores otimizados do problema RDO com $\alpha = 0.8$, exemplo 3.

Descrição	Valores Finais
Var. Projecto 1 (m ²)	0.0678194
Var. Projecto 2 (m ²)	0.0804248
Var. Projecto 3 (m ²)	0.0300052
Var. Projecto 4 (m)	4.87333
Var. Projecto 5 (m)	3.68626
Var. Projecto 6 (m)	5.50725
Var. Projecto 7 (m)	6.00000
Var. Projecto 8 (m)	5.50764
Var. Projecto 9 (m)	3.68534
Var. Projecto 10 (m)	4.87344
Volume (m ³)	6.04041
Variância	0.066526
β_1	5.66376
β_2	7.52037
β_3	6.34130
β_4	6.42718
β_5	6.34164
β_6	6.42736
β_7	7.52360

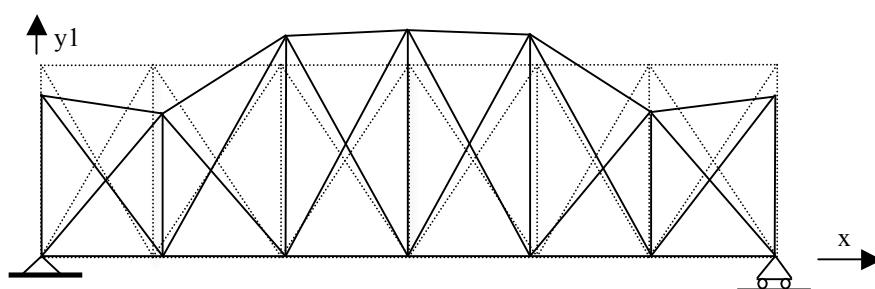


Figura 6.11 - Configuração final sobreposta à inicial, com a formulação RDO e $\alpha = 0.8$ (exemplo 3).

É notório que a configuração final na Figura 6.11 é diferente do obtido com $\alpha = 0$, na Figura 6.8.

A Figura 6.12 representa a evolução dos primeiros quatro valores de β em função do parâmetro α . As Figuras 6.13a) e 6.13b) mostram, respectivamente, o volume e a variância em função de α .

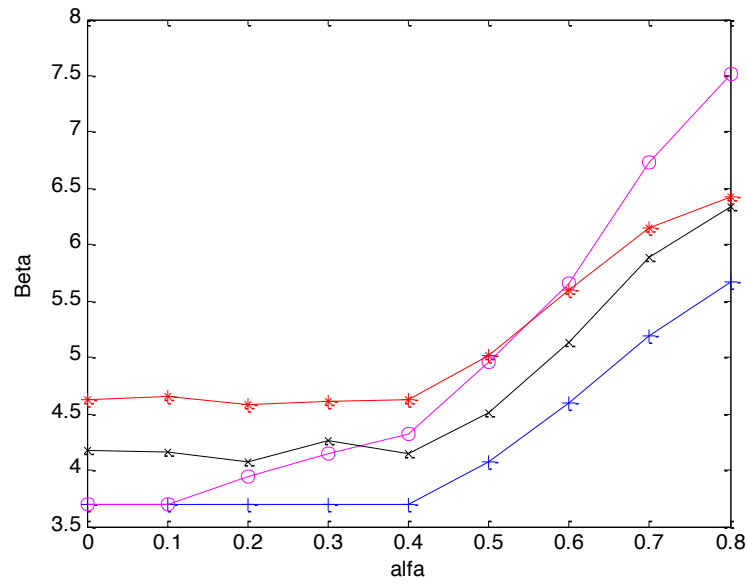


Figura 6.12 - Gráfico de β_1 (azul, +), β_2 (magenta, o), β_3 (preto, x), β_4 (vermelho, *) em função de α , exemplo 3.

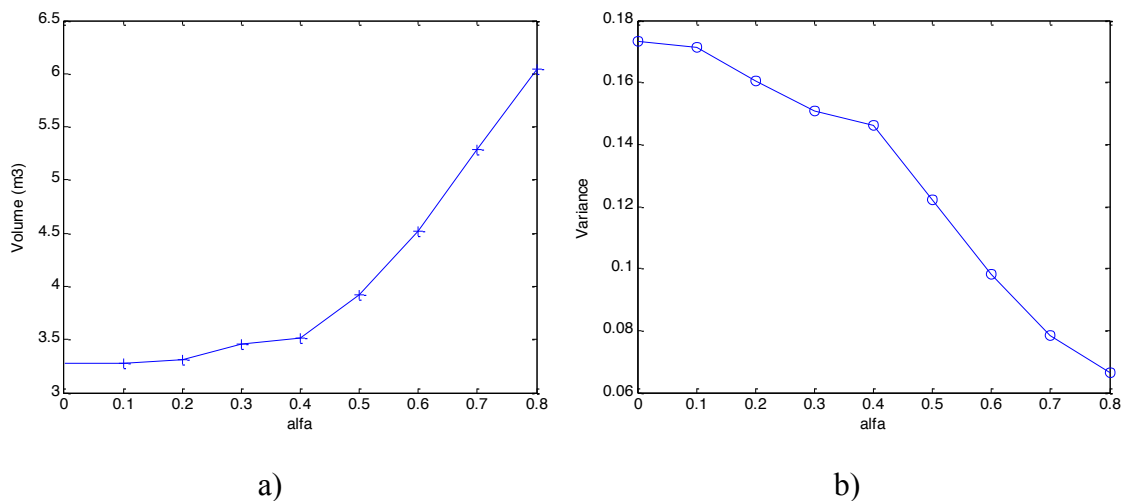


Figura 6.13 - Volume (a) e Variância (b) em função de α , exemplo 3.

Observando a Figura 6.12, é possível ver que β_1 é sempre o índice de fiabilidade mais crítico ou menor. Com as informações fornecidas pela Figura 6.10 e Tabela 6.17 a solução mais adequada pode ser identificada e escolhida. Olhando as Figuras 6.13a) e 6.13b), onde a variação do volume e variância em função de α é traçado, a escolha mais viável é $\alpha = 0.4$.

6.5 Exemplo 4 – Plataforma Offshore

Neste exemplo efectuar-se-á RBDO e RDO numa estrutura 3D, composta por 139 elementos. A estrutura é mostrada na Figura 6.14, e é constituída por seis níveis. Pretende representar uma torre de exploração petrolífera instalada no mar e foi apresentada por Stocki et al. [56].

Os elementos viga 3D são agrupados em 6 conjuntos, e cada um desses conjuntos corresponde a uma secção transversal.

O modelo estocástico do problema é apresentado detalhadamente na Tabela 6.19. Neste caso, foram utilizadas 11 variáveis básicas. Estas incluem as áreas das secções transversais de cada grupo, módulo de Young e o factor de carga.

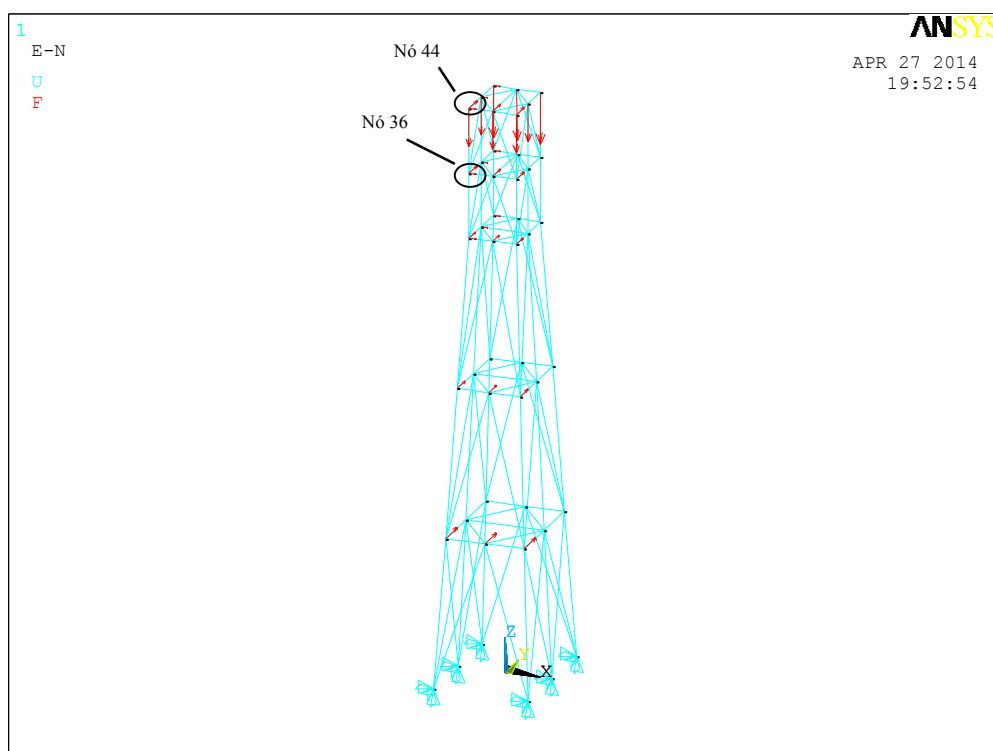


Figura 6.14 - Configuração do exemplo 4.

A estrutura é fixada em 6 apoios simples contrangidos em todas as direcções. As dimensões da base de fixação e do topo são 22 m X 16 m e 11 m X 8 m, respectivamente. A estrutura é composta por 6 andares, cada secção vertical da estrutura até ao 3º andar, inclusive, tem 25 m de altura; a partir do 3º andar para cima cada secção tem 11m de altura, resultando uma altura total de 97m da estrutura.

Os elementos estruturais são divididos em 6 grupos. O primeiro grupo é constituído pelos elementos externos do 1º, 2º e 3º andar da plataforma. No segundo grupo são assumidos os elementos que não fazem parte dos cantos até ao 3º andar. O terceiro grupo é composto pelos elementos diagonais até ao 3º andar, pelos elementos internos do 1º, 2º e do 3º andar e os elementos que não se encontram como canto do 3º andar para cima. O quarto grupo é constituído pelos elementos diagonais a partir do 3º andar e também todos os elementos dos andares 4 e 5. O quinto grupo é formado pelos elementos de canto até ao 3º andar. Finalmente, o sexto grupo é composto pelos elementos de canto do 3º andar para cima [56]. Em anexo [Anexo 5], está apresentado a definição de elementos no ficheiro APDL.

As cargas aplicadas estão divididas em quatro grupos. Os valores das forças aplicadas do grupo 1 são $P_1 = 4500\text{kN}$, $P_3 = 2500\text{kN}$, $P_2 = P_4 = P_5 = 3000\text{kN}$, respectivamente. No segundo grupo, os carregamentos têm o valor de $P_6 = 5500\text{kN}$. O terceiro grupo é composto pelo carregamento $P_7 = 2000\text{kN}$, por fim, o quarto grupo das cargas aplicadas é constituído por $P_8 = 1000\text{kN}$.

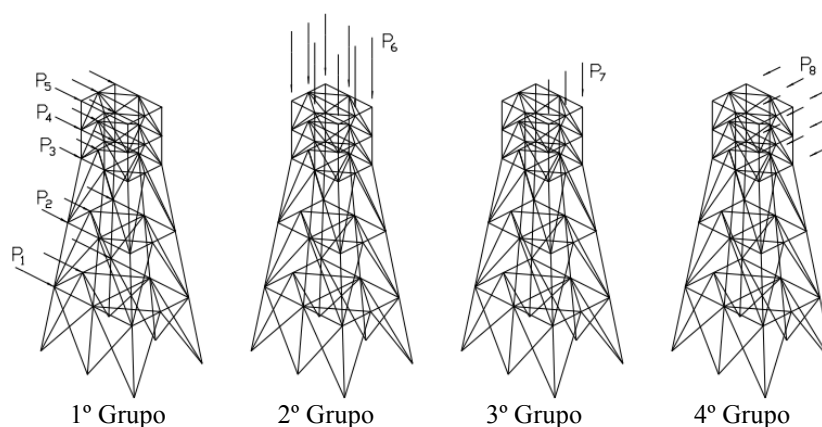


Figura 6.15 - Representação das cargas aplicadas [56].

Tabela 6.19 - Modelo estocástico do exemplo 4.

Variável	Descrição	Distribuição	Média	Desvio Padrão
1	Área da secção transversal grupo 1	Lognormal	0.113391 m ²	0.0113391 m ²
2	Área da secção transversal grupo 2	Lognormal	0.113391 m ²	0.0113391 m ²
3	Área da secção transversal grupo 3	Lognormal	0.113391 m ²	0.0113391 m ²
4	Área da secção transversal grupo 4	Lognormal	0.113391 m ²	0.0113391 m ²
5	Área da secção transversal grupo 5	Lognormal	0.113391 m ²	0.0113391 m ²
6	Área da secção transversal grupo 6	Lognormal	0.113391 m ²	0.0113391 m ²
7	Factor de carga 1	Gumbel	1.0	0.15
8	Factor de carga 2	Lognormal	1.0	0.10
9	Factor de carga 3	Lognormal	1.0	0.20
10	Factor de carga 4	Gumbel	1.0	0.20
11	Módulo Young	Lognormal	210 GPa	10.5 GPa

Em seguida definiu-se o conjunto de performances. Neste exemplo, o volume total é considerado um desempenho estrutural fundamental que deve ser otimizado. Os deslocamentos segundo *Y* dos nós 44 e 36 (representados na Fig. 6.14) também devem ser calculados. A Tabela 6.20 mostra todas as performances definidas, com os valores iniciais correspondentes (calculados pelo PROAES_3D).

O programa desenvolvido deve calcular as derivadas de cada uma das performances na Tabela 6.20 em relação às variáveis da Tabela 6.19.

Tabela 6.20 - Performances do exemplo 4.

Performance	Descrição	Valor inicial
1	Volume	222.15838 m ³
2	Deslocamento do nó 44 segundo Y	+0.619976 m
3	Deslocamento do nó 36 segundo Y	+0.528241 m

Por forma a confirmar a consistência dos cálculos, os valores iniciais das performances foram também calculados pelo programa ANSYS, Tabela 6.21.

Tabela 6.21 - Performances do exemplo 4 (calculadas pelo programa ANSYS).

Performance	Descrição	Valor inicial
1	Volume	222,15838 m ³
2	Deslocamento do nó 44 segundo Y	+0,61998 m
3	Deslocamento do nó 36 segundo Y	+0,52824 m

Calcularam-se as sensibilidades através do programa PROAES_3D e também por diferenças finitas, como mostram as Tabelas 6.22 e 6.23, respectivamente.

Tabela 6.22 - Sensibilidades do exemplo 4 (calculadas pelo PROAES_3D).

Variável\Perf.	1	2	3
1	$1.520000 \times 10^{+2}$	-4.288871×10^{-2}	-4.111954×10^{-2}
2	$7.385379 \times 10^{+2}$	-1.758399	-1.652969
3	$4.830495 \times 10^{+2}$	-9.281519×10^{-2}	-5.657211×10^{-2}
4	$1.964059 \times 10^{+2}$	-8.676765×10^{-3}	-5.550731×10^{-3}
5	$3.012308 \times 10^{+2}$	-2.858877	-2.318266
6	$8.800000 \times 10^{+1}$	-2.865453×10^{-2}	-3.929948×10^{-3}
7	0.0	$+6.067556 \times 10^{-1}$	$+5.197385 \times 10^{-1}$
8	0.0	-3.946933×10^{-5}	-5.551555×10^{-5}
9	0.0	$+1.324617 \times 10^{-2}$	$+8.572709 \times 10^{-3}$
10	0.0	$+1.406183 \times 10^{-5}$	-1.498045×10^{-5}
11	0.0	$-2.952268 \times 10^{-12}$	$-2.515432 \times 10^{-12}$

Tabela 6.23 - Sensibilidades do exemplo 4 (calculadas pelo método das diferenças finitas, com $\Delta = 0.1\%$).

Variável\Perf.	1	2	3
1	$1.520000 \times 10^{+2}$	-4.285104×10^{-2}	-4.108347×10^{-2}
2	$7.385379 \times 10^{+2}$	-1.75726	-1.651867
3	$4.830495 \times 10^{+2}$	-9.274663×10^{-2}	-5.653134×10^{-2}
4	$1.964059 \times 10^{+2}$	-8.668574×10^{-3}	-5.545868×10^{-3}
5	$3.012308 \times 10^{+2}$	-2.857103	-2.316829
6	$8.800000 \times 10^{+1}$	-2.863549×10^{-2}	-3.927398×10^{-3}
7	0.0	$+6.067556 \times 10^{-1}$	$+5.197385 \times 10^{-1}$
8	0.0	-3.946940×10^{-5}	-5.551560×10^{-5}
9	0.0	$+1.324617 \times 10^{-2}$	$+8.572709 \times 10^{-3}$
10	0.0	$+1.406180 \times 10^{-5}$	-1.498040×10^{-5}
11	0.0	$-2.949319 \times 10^{-12}$	$-2.512919 \times 10^{-12}$

Associadas às performances 2 e 3, foram estabelecidas duas funções de estado limite (LSF). A Tabela 6.24 mostra os índices de fiabilidade (calculados pelo FORM) de cada LSF. As duas LSF são definidas como $g(X) = \delta_{Adm} - \delta_{Nó}$.

Considerou-se o deslocamento admissível $\delta_{Adm} = 0.9 \text{ m}$ para a primeira e $\delta_{Adm} = 0.8 \text{ m}$ para a segunda.

Tabela 6.24 - LSF e índices de fiabilidade do exemplo 4.

LSF	Performance	Descrição	β
1	2	Deslocamento do nó 44 segundo Y	2.1196
2	3	Deslocamento do nó 36 segundo Y	2.3074

Nas Tabelas 6.25 e 6.26 estão apresentadas as derivadas de β_1 em ordem aos valores médios e desvios padrão das variáveis.

Tabela 6.25 - Derivadas de β_1 em ordem aos valores médios das variáveis, exemplo 4.

Variável	$d\beta / d\mu_{(x)}(\text{PROAES_FORM})$	$d\beta / d\mu_{(x)}(\text{diferenças finitas})$
1	3.185376×10^{-1}	3.182581×10^{-1}
2	$1.394602 \times 10^{+1}$	$1.393792 \times 10^{+1}$
3	6.750194×10^{-1}	6.745227×10^{-1}
4	6.382461×10^{-2}	6.376432×10^{-2}
5	$2.244581 \times 10^{+1}$	$2.243457 \times 10^{+1}$
6	1.974970×10^{-1}	1.973666×10^{-1}
7	-3.345884	-3.347639
8	2.118735×10^{-4}	2.119000×10^{-4}
9	-7.438870×10^{-2}	-7.439030×10^{-2}
10	-7.531883×10^{-5}	-7.530000×10^{-5}
11	2.270082×10^{-11}	2.268340×10^{-11}

Tabela 6.26 - Derivadas de β_1 em ordem aos valores desvios padrão das variáveis, exemplo 4.

Variável	$d\beta / d\sigma_{(x)}(\text{PROAES_FORM})$	$d\beta / d\sigma_{(x)}(\text{diferenças finitas})$
1	-3.358516×10^{-2}	-3.358291×10^{-2}
2	-5.590494	-5.592710
3	-7.670460×10^{-2}	-7.670803×10^{-2}
4	-6.354464×10^{-3}	-6.349710×10^{-3}
5	$-1.276101 \times 10^{+1}$	$-1.276589 \times 10^{+1}$
6	-2.027224×10^{-2}	-2.027498×10^{-2}
7	-8.159521	-8.154173
8	-2.079123×10^{-5}	-2.099999×10^{-5}
9	1.174504×10^{-2}	1.174450×10^{-2}
10	1.237176×10^{-5}	1.250000×10^{-5}
11	$-1.194876 \times 10^{-11}$	$-1.195393 \times 10^{-11}$

Tendo estabelecido o modelo estocástico, performances e LSF, um problema RBDO clássico é então formulado.

O volume total é a função objectivo a ser minimizada com restrições sobre os índices de fiabilidade, todos devem ser maiores do que $\beta_{Min} = 3.7$. Um total de 6 variáveis de projeto são utilizadas, estão associadas aos valores médios das variáveis básicas 1 a 6 da Tabela 6.19, que corresponde às áreas das secções transversais. Todas as variáveis de projeto são apresentadas na Tabela 6.27, com os valores iniciais e limites inferior e superior.

A fim de resolver o problema com os programas desenvolvidos, os ficheiros de entrada que descrevem a análise de elementos finitos e o modelo estocástico são criados.

Também devem ser desenvolvidas duas funções MATLAB, para calcular a função objectivo e os constrangimentos. A *optimtool* do MATLAB é usada com o algoritmo de programação quadrática sequencial (SQP).

Tabela 6.27 - Variáveis de projecto para as formulações RBDO e RDO do exemplo 4.

Variável Projecto	Descrição	Valor inicial	Limite inferior	Limite superior
1	Área da secção transversal grupo 1	0.113391 m ²	0.04082 m ²	0.70869 m ²
2	Área da secção transversal grupo 2	0.113391 m ²	0.04082 m ²	0.70869 m ²
3	Área da secção transversal grupo 3	0.113391 m ²	0.04082 m ²	0.70869 m ²
4	Área da secção transversal grupo 4	0.113391 m ²	0.04082 m ²	0.70869 m ²
5	Área da secção transversal grupo 5	0.113391 m ²	0.04082 m ²	0.70869 m ²
6	Área da secção transversal grupo 6	0.113391 m ²	0.04082 m ²	0.70869 m ²

6.5.1 RBDO

A convergência é obtida em 6 iterações, como mostra a Figura 6.15.

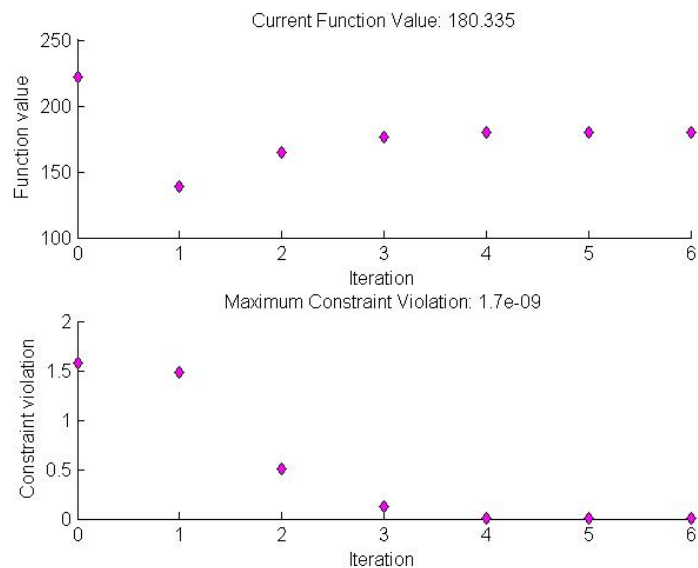


Figura 6.16 - Convergência para o problema RBDO do exemplo 4.

A Tabela 6.28 apresenta os valores iniciais e os valores finais obtidos das variáveis de projecto, da função objectivo e dos constrangimentos.

Para este problema, o tempo total de optimização é de apenas 52 s e requer 21 chamadas para que o programa calcule a função objectivo, constrangimentos e gradientes.

Tabela 6.28 - Valores iniciais e finais para o problema RBDO do exemplo 4.

Descrição	Inicial	Final
Var. Projecto 1 (m ²)	0.113391	0.040820
Var. Projecto 2 (m ²)	0.113391	0.074873
Var. Projecto 3 (m ²)	0.113391	0.040820
Var. Projecto 4 (m ²)	0.113391	0.040820
Var. Projecto 5 (m ²)	0.113391	0.290496
Var. Projecto 6 (m ²)	0.113391	0.040820
Volume (m ³)	222.15838	180.33464
β_1	2.1196	3.7000
β_2	2.3074	3.7000

6.5.2 RDO

Neste caso, considerou-se a primeira LSF crítica e a variância do deslocamento, δ , do nó 44 segundo Y calculada pela equação (6.3). Esta LSF foi escolhido por estar sempre activa durante o processo de optimização e por ser considerada como uma restrição crítica durante o tempo de vida da estrutura.

$$\text{var } \delta = \sum_{i=1}^{NV} \left(\frac{\partial \delta}{\partial x_i} \right)^2 \cdot \sigma_{x_i}^2 \quad (6.3)$$

O desvio padrão é mantido constante e igual ao inicial, Tabela 6.19.

Uma estrutura optimizada deve ter baixo peso e alta fiabilidade. Mas é requerido que mantenha um nível de segurança constante na presença de acções aleatórias.

Como já mencionado anteriormente, o conceito de robustez está associado à menor variabilidade da resposta estrutural no que diz respeito às incertezas do projeto. Formulou-se o problema RDO, onde a função objectivo é tida como uma soma ponderada do volume estrutural e da variância de δ , equação (6.4).

$$f = (1 - \alpha) \frac{\text{vol}}{c_1} + \alpha \frac{\text{var}[\delta]}{c_2} \quad (6.4)$$

onde vol e $\text{var}[\delta]$ representam o volume total e a variância de δ , respectivamente, c_1 e c_2 são as constantes que normalizam os dois termos da equação e α é um coeficiente que permite definir a importância relative desses termos. Para o presente caso $c_1 = 200$ e $c_2 = 0.005$.

Variando-se o valor de α no intervalo $[0;0.8]$, é possível obter a frente de Pareto, Figura 6.16, de soluções não-denominadas óptimas mostradas na Tabela 6.29.

Tabela 6.29 - Volume e Variância em função de α , exemplo 4.

α	Volume(m ³)	Variância
0	180.334642	0.004989
0,1	180.334642	0.004989
0,2	180.334642	0.004989
0,3	184.386850	0.004742
0,4	205.513968	0.003797
0,5	233.851031	0.002844
0,6	264.029634	0.002227
0,7	303.281807	0.001703
0,8	363.697118	0.001207

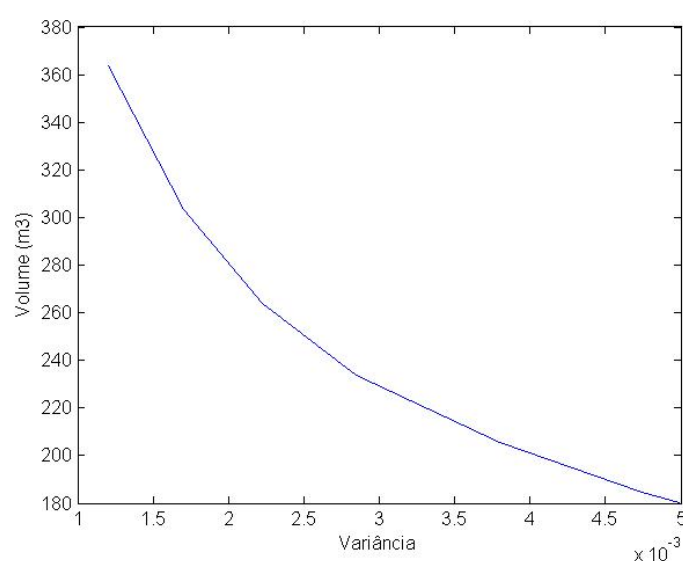


Figura 6.17 - Frente de Parteo das soluções ótimas para a RDO, exemplo 4.

Claramente menores valores de variância não têm interesse para o estudo, pois estão associados a volumes estruturais muito elevados. Como os constrangimentos impostos foram apenas nos limites inferiores de β , as soluções otimizadas para valores elevados de α são muito seguras, com um alto índice de fiabilidade, como mostra a Tabela 6.30, em que a solução otimizada para $\alpha = 0.8$ é apresentada.

Tabela 6.30 - Valores otimizados do problema RDO com $\alpha = 0.8$, exemplo 4.

Descrição	Valor Final
Var. Projecto 1 (m ²)	0.043218
Var. Projecto 2 (m ²)	0.156151
Var. Projecto 3 (m ²)	0.055501
Var. Projecto 4 (m)	0.040820
Var. Projecto 5 (m)	0.658496
Var. Projecto 6 (m)	0.097942
Volume (m ³)	363.697118
Variância	0.001207
β_1	6.704415
β_2	6.704415

6.6 Conclusões

Neste capítulo apresentaram-se os resultados dos exemplos propostos, efectuou-se análise de elementos finitos e/ou sensibilidades, análise de fiabilidade, RBDO e RDO.

Os dois primeiros exemplos foram utilizados para demonstrar a precisão e facilidade com que os programas desenvolvidos efectuam a análise de elementos finitos, de sensibilidades e de fiabilidade. Efectuou-se também uma comparação das sensibilidades calculadas pelo método contínuo e pelo método das diferenças finitas, bem como uma comparação dos resultados da análise de fiabilidade do exemplo 2, obtidos pelos programas desenvolvidos, com os resultados do FERUM.

Em seguida, utilizou-se uma estrutura de treliça e uma plataforma para demonstrar as capacidades dos programas desenvolvidos na resolução de problemas de optimização. Começou-se pela RBDO, minimizando o volume e impondo um valor mínimo aos índices de fiabilidade, depois procedeu-se à RDO mantendo-se a função objectivo (volume) e adicionando variância do índice de fiabilidade no exemplo 3 e do deslocamento no exemplo 4. A partir das soluções não-dominadas óptimas obtém-se a frente de Pareto, desta forma percebe-se a evolução da função objectivo e da variância.

Capítulo 7 - Conclusões

O objectivo principal desta dissertação consistiu no desenvolvimento de um conjunto de programas (em ambiente MATLAB), adequados ao projecto óptimo de estruturas. Estes apresentam a capacidade de executar análise de elementos finitos, análise de sensibilidades, análise de fiabilidade utilizando o método FORM e optimização recorrendo ao algoritmo SQP existente na *toolbox* do MATLAB. Esses programas possibilitam ao utilizador a realização de RBDO e RDO.

De modo a validar os resultados obtidos utilizaram-se quatro exemplos, em que as soluções obtidas, quando comparadas com soluções existentes de outros autores, podem ser consideradas coerentes e precisas.

Considera-se por isso que o conjunto de programas desenvolvidos nesta dissertação fornece uma plataforma básica e fiável para ser usada e desenvolvida por estudantes de engenharia estrutural e investigadores, a fim de promover a aplicação destas técnicas de optimização.

Bibliografia

- [1] H.-G. Beyer and B. Sendhoff, “Robust optimization – A comprehensive survey,” *Comput. Methods Appl. Mech. Eng.*, vol. 196, no. 33–34, pp. 3190–3218, Jul. 2007.
- [2] J. C. Maxwell, “I.—On Reciprocal Figures, Frames, and Diagrams of Forces,” *Earth Environ. Sci. Trans. R. Soc. Edinburgh*, vol. 26, no. 01, pp. 1–40, 1870.
- [3] A. G. M. Michell, “LVIII. The limits of economy of material in frame-structures,” *Philos. Mag. Ser. 6*, vol. 8, no. 47, pp. 589–597, Nov. 1904.
- [4] P. G. Coelho and J. B. Cardoso, “Apontamentos de Métodos Computacionais Em Engenharia Mecânica,” *Dep. Eng. Mecânica e Ind. Fac. Ciências e Tecnol. - Univ. Nov. Lisboa*, 2012.
- [5] R. Courant, “Variational Methods for the Solution of Problems of Equilibrium and Vibrations,” *Bull. Am. Math. Soc.*, vol. 49, pp. 1–23, 1942.
- [6] R. W. Clough, “The Finite Element Method in Plane Stress Analysis,” *J. Struct. Div. ASCE, Proc. 2nd Conf. Electron. Comput.*, pp. 345–378, 1960.
- [7] L. A. Schmit, “Structural Design by Systematic Synthesis,” *Proc. 2d Conf. Electron. Comput. ASCE, Pittsburgh, Pa.*, pp. 105–132, 1960.
- [8] M. Kleiber, a. Siemaszko, and R. Stocki, “Interactive stability-oriented reliability-based design optimization,” *Comput. Methods Appl. Mech. Eng.*, vol. 168, no. 1–4, pp. 243–253, Jan. 1999.
- [9] Z. Kang and V. Vorgelegt, “Robust design optimization of structures under uncertainties,” *Ph.D. Thesis, Inst. für Statik und Dynamik der Luft - und Raumfahrkon - struktionen Univ. Stuttgart*, p. 161, 2005.

- [10] Z. Kang and S. Bai, "On robust design optimization of truss structures with bounded uncertainties," *Struct. Multidiscip. Optim.*, vol. 47, no. 5, pp. 699–714, Jan. 2013.
- [11] J. B. Cardoso, A. P. Teixeira, and P. T. Fraga, "Reliability-Based Design Optimization Using Design Sensitivity Analysis," in *7th Int. Conf. on Computational Stochastic Mechanics (CSM7)*, 2014.
- [12] R. Stocki, T. Szolc, P. Tazowski, and J. Knabel, "Robust design optimization of the vibrating rotor-shaft system subjected to selected dynamic constraints," *Mech. Syst. Signal Process.*, vol. 29, pp. 34–44, May 2012.
- [13] A. Der Kiureghian, T. Haukaas, and K. Fujimura, "Structural reliability software at the University of California, Berkeley," *Struct. Saf.*, vol. 28, no. 1–2, pp. 44–67, Jan. 2006.
- [14] R. E. Melchers, *Structural Reliability Analysis and Prediction*, 2nd Edit. Chichester: John Wiley & Sons Ltd, 1999.
- [15] T. Baptista, "Análise de fiabilidade de estruturas," Dissertação de Mestrado, Universidade Nova de Lisboa - Faculdade de Ciências e Tecnologia, 2012.
- [16] M. H. Faber, "Risk and Safety in Civil, Surveying and Environmental Engineering." Swiss Federal Institute of Technology ETH Zurich, Switzerland, 2006.
- [17] C. A. Cornell, "A Probability-Based Structural Code," *ACI J. Proc.*, vol. 66, no. 12, pp. 974–985, 1969.
- [18] M. Freudenthal, A.M.; Garrelts, J.M.; Shinozuka, "The analysis of structural safety," *J. Struct. Div.*, vol. 92, 1966.
- [19] A. M. Hasofer and L. N.C., "An exact and invariant first order reliability format," *J. Eng. Mech. Div.*, vol. 100, pp. 111–121, 1974.
- [20] R. Rackwitz and B. Flessler, "Structural reliability under combined random load sequences," *Comput. Struct.*, vol. 9, no. 5, pp. 489–494, 1978.

- [21] X. Chen and N. C. Lind, "Fast probability integration by three-parameter normal tail approximation," *Struct. Saf.*, vol. 1, no. 4, pp. 269–276, Jan. 1982.
- [22] Y. Wu and P. Wirsching, "New Algorithm for Structural Reliability Estimation," *J. Eng. Mech.*, vol. 113, no. 9, pp. 1319–1336, Sep. 1987.
- [23] P. L. Liu, *Finite-element Reliability Methods for Geometrically Nonlinear Stochastic Structures*. Department of Civil Engineering, University of California, 1989.
- [24] O. Ditlevsen and H. O. Madsen, *Structural Reliability Methods*. Wiley, 1996, p. 384.
- [25] D. P. Rubinstein, R.Y.; Kroese, *Simulation and the Monte Carlo method*, vol. 60, no. 1. New York: John Wiley & Sons, 1981, p. 79.
- [26] E. Vanmarcke and M. Grigoriu, "Stochastic Finite Element Analysis of Simple Beams," *J. Eng. Mech.*, vol. 109, no. 5, pp. 1203–1214, Oct. 1983.
- [27] A. Der Kiureghian and J.-B. Ke, "The stochastic finite element method in structural reliability," *Probabilistic Eng. Mech.*, vol. 3, no. 2, pp. 83–91, 1988.
- [28] P. Spanos and R. Ghanem, "Stochastic Finite Element Expansion for Random Media," *J. Eng. Mech.*, vol. 115, no. 5, pp. 1035–1053, May 1989.
- [29] J. Teigen, D. Frangopol, S. Sture, and C. Felippa, "Probabilistic FEM for Nonlinear Concrete Structures. II: Applications," *J. Struct. Eng.*, vol. 117, no. 9, pp. 2690–2707, Sep. 1991.
- [30] M. Rajashekhar and B. Ellingwood, "Reliability of Reinforced-Concrete Cylindrical Shells," *J. Struct. Eng.*, vol. 121, no. 2, pp. 336–347, Feb. 1995.
- [31] J. Zhang and B. Ellingwood, "SFEM for Reliability of Structures with Material Nonlinearities," *J. Struct. Eng.*, vol. 122, no. 6, pp. 701–704, Jun. 1996.

- [32] F. Duprat, A. Sellier, X. S. Nguyen, and G. Pons, "The projection gradient algorithm with error control for structural reliability," *Eng. Struct.*, vol. 32, no. 11, pp. 3725–3733, 2010.
- [33] T. Abdo and R. Rackwitz, "A New Beta-Point Algorithm for Large Time-Invariant and Time-Variant Reliability Problems," in *Reliability and Optimization of Structural Systems '90 SE - I*, vol. 61, A. Der Kiureghian and P. Thoft-Christensen, Eds. Springer Berlin Heidelberg, 1991, pp. 1–12.
- [34] Y. Zhang and A. Der Kiureghian, "Dynamic response sensitivity of inelastic structures," *Comput. Methods Appl. Mech. Eng.*, vol. 108, no. 1, pp. 23–36, 1993.
- [35] A. Zhang, Y., Der Kiureghin, "Finite element reliability methods for inelastic structures." Dep. Civil and Env. Eng. University of California, Berkeley (CA), 1997.
- [36] J. B. Cardoso, "Optimização de Configuração de Estruturas," Dissertação de Doutorado, Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia, 1994.
- [37] K. K. Choi and E. J. Haug, "Shape Design Sensitivity Analysis of Elastic Structures," *J. Struct. Mech.*, vol. 11, no. 2, pp. 231–269, Jan. 1983.
- [38] V. Komkov, K. K. Choi, and E. J. Haug, *Design Sensitivity Analysis of Structural Systems*. New York, Academic Press, 1986.
- [39] C. A. Mota Soares, "Computer Aided Optimal Design : Structural and Mechanical Systems." Berlin, Heidelberg, New York, Springer, 1987.
- [40] J. D. Sørensen, P. Thoft-Christensen, A. Siemaszko, J. B. Cardoso, and J. L. T. Santos, "Interactive Reliability-Based Structural Design and Optimization," in *6th IFIP WG 7.5 Working Conference, Assisi, Italy*, 1994.
- [41] J. B. Cardoso, J. R. de Almeida, J. M. Dias, and P. G. Coelho, "Structural reliability analysis using Monte Carlo simulation and neural networks," *Adv. Eng. Softw.*, vol. 39, no. 6, pp. 505–513, Jun. 2008.

- [42] I. Kaymaz, C. McMahon, and X. Meng, “Reliability Based Structural Optimizaiton using the Response Surface method and Monte Carlo simulation,” in *8th International Machine Design and Production Conference, Ankara, Turkey*, 1998.
- [43] Y. K. Wen, “Minimum lifecycle cost design under multiple hazards,” *Reliab. Eng. Syst. Saf.*, vol. 73, no. 3, pp. 223–231, Sep. 2001.
- [44] G. I. Schuëller and H. a. Jensen, “Computational methods in optimization considering uncertainties – An overview,” *Comput. Methods Appl. Mech. Eng.*, vol. 198, no. 1, pp. 2–13, Nov. 2008.
- [45] Y. Ben-Haim and I. Elishakof, “Convex Models of Uncertainty in Applied Mechanics,” *Elsevier Sci. Publ. Amsterdam*, 1990.
- [46] C. Jiang, X. Han, G. R. Liu, and G. P. Liu, “A nonlinear interval number programming method for uncertain optimization problems,” *Eur. J. Oper. Res.*, vol. 188, no. 1, pp. 1–13, Jul. 2008.
- [47] N. Impollonia and G. Muscolino, “Interval analysis of structures with uncertain-but-bounded axial stiffness,” *Comput. Methods Appl. Mech. Eng.*, vol. 200, no. 21–22, pp. 1945–1962, May 2011.
- [48] C. Jiang, X. Han, and G. R. Liu, “Optimization of structures with uncertain constraints based on convex model and satisfaction degree of interval,” *Comput. Methods Appl. Mech. Eng.*, vol. 196, no. 49–52, pp. 4791–4800, Nov. 2007.
- [49] C. Jiang, X. Han, G. Y. Lu, J. Liu, Z. Zhang, and Y. C. Bai, “Correlation analysis of non-probabilistic convex model and corresponding structural reliability technique,” *Comput. Methods Appl. Mech. Eng.*, vol. 200, no. 33–36, pp. 2528–2546, Aug. 2011.
- [50] E. Sandgren and T. M. Cameron, “Robust design optimization of structures through consideration of variation,” *Comput. Struct.*, vol. 80, no. 20–21, pp. 1605–1613, Aug. 2002.

- [51] N. D. Lagaros, V. Plevris, and M. Papadrakakis, “Multi-objective design optimization using cascade evolutionary computations,” *Comput. Methods Appl. Mech. Eng.*, vol. 194, no. 30–33, pp. 3496–3515, Aug. 2005.
- [52] I. Doltsinis and Z. Kang, “Perturbation-based stochastic FE analysis and robust design of inelastic deformation processes,” *Comput. Methods Appl. Mech. Eng.*, vol. 195, no. 19–22, pp. 2231–2251, Apr. 2006.
- [53] J. K. Guest and T. Igusa, “Structural optimization under uncertain loads and nodal locations,” *Comput. Methods Appl. Mech. Eng.*, vol. 198, no. 1, pp. 116–124, Nov. 2008.
- [54] L. Harzheim and U. Warnecke, “Robustness optimization of the position of an anti-roll bar link to avoid the toggling of a rear axle stabilizer,” *Struct. Multidiscip. Optim.*, vol. 42, no. 2, pp. 315–323, Mar. 2010.
- [55] S. Chen and W. Chen, “A new level-set based approach to shape and topology optimization under geometric uncertainty,” *Struct. Multidiscip. Optim.*, vol. 44, no. 1, pp. 1–18, May 2011.
- [56] R. Stocki, A. Siemaszko, and M. Kleiber, “Interactive methodology for reliability-based structural design and optimization,” *Comput. Assist. Mech. Eng. Sci.*, no. 6, pp. 39–62, 1999.

Anexos

Anexo 1:

Ficheiro de dados **portico.inp**, do exemplo 1

```
#
# Programa PROAES
#
# Exemplo do pórtico em Cardoso [36]
#
# Definição dos nós
#
n, 1, 0, 0
n, 2, 0, 2
n, 3, 2, 1
n, 4, 2, 0
#
# Elementos
#
e, 1, 3, 1, 1, 1, 2
e, 2, 3, 1, 1, 2, 3
e, 3, 3, 1, 1, 3, 4
#
# Material
#
m, 1, 210e9
#
# Características da secção
#
s, 1, 3.14159e-4, 7.85398e-9, 0.02
#
# Factor de carga
#
k, 1, 1
#
# Cargas concentradas aplicadas proporcionais a k
#
z, 3, 1, 200, 0, 0
#
# Apoios
#
a, 1, 1, 0
a, 1, 2, 0
a, 1, 6, 0
a, 4, 1, 0
a, 4, 2, 0
a, 4, 6, 0
#
# Variáveis de projecto
#
v, 1, 1, 2, 1
v, 2, 1, 2, 2
v, 3, 2, 1
v, 4, 3, 1
v, 5, 4, 1
v, 6, 5, 1
#
# Performances
#
p, 1, 2, 2, 1
p, 2, 2, 3, 1
p, 3, 2, 2, 2
p, 4, 2, 3, 2
p, 5, 1
# (fim do ficheiro)
```

Anexo 2:

Ficheiro **opt2ef** do exemplo 1

```
3
6
1, 0.0
2, 2.0
3, 210e9
4, 1.0
5, 3.14159e-4
6, 7.85398e-9
5
1
2
3
4
5
```

Anexo 3:

Ficheiro de dados **portico_2andares.inp**, exemplo 2.

```
#
# Programa PROAES
#
# Exemplo do portico de 2 andares, manual do programa FERUM [13]
#
# Definição dos nós
#
n, 1, 0, 0
n, 2, 7500, 0
n, 3, 15000, 0
n, 4, 0, 3000
n, 5, 7500, 3000
n, 6, 15000, 3000
n, 7, 0, 6000
n, 8, 7500, 6000
n, 9, 15000, 6000
#
# Elementos
#
e, 1, 3, 1, 1, 1, 4
e, 2, 3, 2, 2, 2, 5
e, 3, 3, 3, 3, 3, 6
e, 4, 3, 4, 4, 4, 5
e, 5, 3, 5, 5, 5, 6
e, 6, 3, 6, 6, 4, 7
e, 7, 3, 7, 7, 5, 8
e, 8, 3, 8, 8, 6, 9
e, 9, 3, 9, 9, 7, 8
e, 10, 3, 10, 10, 8, 9
#
# Material
#
m, 1, 200000
m, 2, 200000
m, 3, 200000
m, 4, 200000
m, 5, 200000
m, 6, 200000
m, 7, 200000
m, 8, 200000
m, 9, 200000
m, 10, 200000
#
# Caraterísticas da secção
#
s, 1, 5000, 0.5e9
s, 2, 5000, 0.5e9
s, 3, 5000, 0.5e9
s, 4, 5000, 0.5e9
s, 5, 5000, 0.5e9
s, 6, 5000, 0.5e9
s, 7, 5000, 0.5e9
s, 8, 5000, 0.5e9
s, 9, 5000, 0.5e9
s, 10, 5000, 0.5e9
#
# Factor de carga
#
k, 1, 300000
#
# Cargas concentradas aplicadas proporcionais a k
#
z, 4, 1, 1, 0, 0
z, 7, 1, 2, 0, 0
#
```

```

#
# Apoios
#
a, 1, 1, 0
a, 1, 2, 0
a, 1, 6, 0
a, 2, 1, 0
a, 2, 2, 0
a, 2, 6, 0
a, 3, 1, 0
a, 3, 2, 0
a, 3, 6, 0
#
# Variáveis de projecto
#
v, 1, 2, 1
v, 2, 2, 2
v, 3, 2, 3
v, 4, 2, 4
v, 5, 2, 5
v, 6, 2, 6
v, 7, 2, 7
v, 8, 2, 8
v, 9, 2, 9
v, 10, 2, 10
v, 11, 5, 1
v, 12, 5, 2
v, 13, 5, 3
v, 14, 5, 4
v, 15, 5, 5
v, 16, 5, 6
v, 17, 5, 7
v, 18, 5, 8
v, 19, 5, 9
v, 20, 5, 10
v, 21, 4, 1
v, 22, 4, 2
v, 23, 4, 3
v, 24, 4, 4
v, 25, 4, 5
v, 26, 4, 6
v, 27, 4, 7
v, 28, 4, 8
v, 29, 4, 9
v, 30, 4, 10
v, 31, 3, 1
#
# Performances
#
p, 1, 1
p, 2, 2, 9, 1
#
# (fim do ficheiro)

```

Anexo 4:

Ficheiro de dados **portico_2andares.frm**, exemplo 2.

```
31
1 2 200000 20000
2 2 200000 20000
3 2 200000 20000
4 2 200000 20000
5 2 200000 20000
6 2 200000 20000
7 2 200000 20000
8 2 200000 20000
9 2 200000 20000
10 2 200000 20000
11 2 0.5E+9 0.5E+8
12 2 0.5E+9 0.5E+8
13 2 0.5E+9 0.5E+8
14 2 0.5E+9 0.5E+8
15 2 0.5E+9 0.5E+8
16 2 0.5E+9 0.5E+8
17 2 0.5E+9 0.5E+8
18 2 0.5E+9 0.5E+8
19 2 0.5E+9 0.5E+8
20 2 0.5E+9 0.5E+8
21 2 5000 500
22 2 5000 500
23 2 5000 500
24 2 5000 500
25 2 5000 500
26 2 5000 500
27 2 5000 500
28 2 5000 500
29 2 5000 500
30 2 5000 500
31 3 300000 45000
0
1
2 1 45.0
```

Anexo 5:

Definição do modelo no ficheiro APDL do exemplo 4.

```
! Exemplo OFFSHORE JACKET

/PREP7

! Number of nodes = 46
! Fixation nodes

n, 1, 8, -11, 0
n, 2, 8, 0, 0
n, 3, 8, 11, 0
n, 4, -8, -11, 0
n, 5, -8, 0, 0
n, 6, -8, 11, 0

! 1st deck nodes

n, 7, 6.667, -9.167, 25
n, 8, 6.667, 0.0, 25
n, 9, 6.667, 9.167, 25
n, 10, 0.0, -9.167, 25
n, 11, 0.0, 9.167, 25
n, 12, -6.667, -9.167, 25
n, 13, -6.667, 0.0, 25
n, 14, -6.667, 9.167, 25

! 2nd deck nodes

n, 15, 5.333, -7.333, 50
n, 16, 5.333, 0.0, 50
n, 17, 5.333, 7.333, 50
n, 18, 0.0, -7.333, 50
n, 19, 0.0, 7.333, 50
n, 20, -5.333, -7.333, 50
n, 21, -5.333, 0.0, 50
n, 22, -5.333, 7.333, 50

! 3rd deck nodes

n, 23, 4.0, -5.5, 75
n, 24, 4.0, 0.0, 75
n, 25, 4.0, 5.5, 75
n, 26, 0.0, -5.5, 75
n, 27, 0.0, 5.5, 75
n, 28, -4.0, -5.5, 75
n, 29, -4.0, 0.0, 75
n, 30, -4.0, 5.5, 75

! 4th deck nodes

n, 31, 4.0, -5.5, 86
n, 32, 4.0, 0.0, 86
n, 33, 4.0, 5.5, 86
n, 34, 0.0, -5.5, 86
n, 35, 0.0, 5.5, 86
n, 36, -4.0, -5.5, 86
n, 37, -4.0, 0.0, 86
n, 38, -4.0, 5.5, 86

! 5th deck nodes

n, 39, 4.0, -5.5, 97
n, 40, 4.0, 0.0, 97

n, 41, 4.0, 5.5, 97
n, 42, 0.0, -5.5, 97
n, 43, 0.0, 5.5, 97
n, 44, -4.0, -5.5, 97
n, 45, -4.0, 0.0, 97
n, 46, -4.0, 5.5, 97
```

```

! Elementos

ET,1,BEAM4

! Number of sections = 6

! Area of section: A;Torsion/Momennts of Inertia of sections
! A,Iz,Iy,Jo,height,width

R, 1, 0.1136, 0.06056263, 0.06056263, 0.3, 0.3
R, 2, 0.1136, 0.06056263, 0.06056263, 0.3, 0.3

! diameter=2.083m thick=17.5mm

R, 3, 0.1136, 0.06056263, 0.06056263, 0.3, 0.3
R, 4, 0.1136, 0.06056263, 0.06056263, 0.3, 0.3
R, 5, 0.1136, 0.06056263, 0.06056263, 0.3, 0.3
R, 6, 0.1136, 0.06056263, 0.06056263, 0.3, 0.3

MP,EX,1,210e9
MP,PRXY,1,0.3

! Number of elements = 139
! 1st level elements
! Legs

REAL, 5
en, 1, 1, 7
REAL, 2
en, 2, 1, 8
en, 3, 1, 10
en, 4, 2, 7
REAL, 3
en, 5, 2, 8
REAL, 2
en, 6, 2, 9
en, 7, 3, 8
REAL, 5
en, 8, 3, 9
REAL, 2
en, 9, 3, 11
en, 10, 4, 10
REAL, 5
en, 11, 4, 12
REAL, 2
en, 12, 4, 13
en, 13, 5, 12
REAL, 3
en, 14, 5, 13
REAL, 2
en, 15, 5, 14
en, 16, 6, 13
REAL, 5
en, 17, 6, 14
REAL, 2
en, 18, 6, 11

! Platform
! Outer square

REAL, 1
en, 19, 7, 8
en, 20, 8, 9
en, 21, 9, 11
en, 22, 11, 14
en, 23, 14, 13
en, 24, 13, 12
en, 25, 12, 10
en, 26, 10, 7

! Inner triangles

REAL, 3
en, 27, 10, 8
en, 28, 8, 11
en, 29, 11, 13

```

```

en, 30, 13, 10
en, 31, 8, 13

! 2nd level elements
! Legs

REAL, 5
en, 32, 7, 15
REAL, 2
en, 33, 7, 16
en, 34, 7, 18
REAL, 3
en, 35, 8, 16
REAL, 2
en, 36, 9, 16
REAL, 5
en, 37, 9, 17
REAL, 2
en, 38, 9, 19
en, 39, 12, 18
REAL, 5
en, 40, 12, 20
REAL, 2
en, 41, 12, 21
REAL, 3
en, 42, 13, 21
REAL, 2
en, 43, 14, 21
REAL, 5
en, 44, 14, 22
REAL, 2
en, 45, 14, 19

! Platform
! Outer square

REAL, 1
en, 46, 15, 16
en, 47, 16, 17
en, 48, 17, 19
en, 49, 19, 22
en, 50, 22, 21
en, 51, 21, 20
en, 52, 20, 18
en, 53, 18, 15

! Iner triangles

REAL, 3
en, 54, 18, 16
en, 55, 16, 19
en, 56, 19, 21
en, 57, 21, 18
en, 58, 16, 21

! 3rd level elements
! Legs

REAL, 5
en, 59, 15, 23
REAL, 2
en, 60, 15, 24
en, 61, 15, 26
REAL, 3
en, 62, 16, 24
REAL, 2
en, 63, 17, 24
REAL, 5
en, 64, 17, 25
REAL, 2
en, 65, 17, 27
en, 66, 20, 26
REAL, 5
en, 67, 20, 28
REAL, 2
en, 68, 20, 29
REAL, 3

```



```

en, 69, 21, 29
REAL, 2
en, 70, 22, 29
REAL, 5
en, 71, 22, 30
REAL, 2
en, 72, 22, 27

! Platform
! Outer square

REAL, 1
en, 73, 23, 24
en, 74, 24, 25
en, 75, 25, 27
en, 76, 27, 30
en, 77, 30, 29
en, 78, 29, 28
en, 79, 28, 26
en, 80, 26, 23

! Inner triangles

REAL, 3
en, 81, 26, 24
en, 82, 24, 27
en, 83, 27, 29
en, 84, 29, 26
en, 85, 24, 29

! 4th level elements
! Legs

REAL, 6
en, 86, 23, 31
REAL, 3
en, 87, 23, 32
en, 88, 23, 34
REAL, 4
en, 89, 24, 32
REAL, 3
en, 90, 25, 32
REAL, 6
en, 91, 25, 33
REAL, 3
en, 92, 25, 35
en, 93, 28, 34
REAL, 6
en, 94, 28, 36
REAL, 3
en, 95, 28, 37
REAL, 4
en, 96, 29, 37
REAL, 3
en, 97, 30, 37
REAL, 6
en, 98, 30, 38
REAL, 3
en, 99, 30, 35

! Platform
! Outer square

REAL, 4
en, 100, 31, 32
en, 101, 32, 33
en, 102, 33, 35
en, 103, 35, 38
en, 104, 38, 37
en, 105, 37, 36
en, 106, 36, 34
en, 107, 34, 31

```

```

! Iner triangles

REAL, 4
en, 108, 34, 32
en, 109, 32, 35
en, 110, 35, 37
en, 111, 37, 34
en, 112, 34, 35

! 5th level elements
! Legs

REAL, 6
en, 113, 31, 39
REAL, 3
en, 114, 31, 40
en, 115, 31, 42
REAL, 4
en, 116, 32, 40
REAL, 3
en, 117, 33, 40
REAL, 6
en, 118, 33, 41
REAL, 3
en, 119, 33, 43
en, 120, 36, 42
REAL, 6
en, 121, 36, 44
REAL, 3
en, 122, 36, 45
REAL, 4
en, 123, 37, 45
REAL, 3
en, 124, 38, 45
REAL, 6
en, 125, 38, 46
REAL, 3
en, 126, 38, 43

! Platform
! Outer square

REAL, 4
en, 127, 39, 40
en, 128, 40, 41
en, 129, 41, 43
en, 130, 43, 46
en, 131, 46, 45
en, 132, 45, 44
en, 133, 44, 42
en, 134, 42, 39

! Iner triangles

REAL, 4
en, 135, 42, 40
en, 136, 40, 43
en, 137, 43, 45
en, 138, 45, 42
en, 139, 42, 43

```